Information flow in cybernetic systems.

Clifton, K

INFORMATION FLOW IN CYBERNETIC SYSTEMS

A Thesis Submitted for the Degree of
Doctor of Philosophy in the Faculty
of Science, University of London.

by Kevin Clifton

Electronics Department,
Chelsea College,
London, SW3.

ABSTRACT

The purpose of this paper is to present an account of an investigation into information flow in cybernetic systems. A theory is proposed that brings together information flow and holograms and which establishes a connection between the exponential components of a time domain waveform and the coefficients of the Kolmogorov polynomial. This theory is non-probabilistic and is based upon information concepts borrowed from physical optics.

Computer simulations of Gabor's learning machine, which are based on Kolmogorov's polynomial, are used to investigate the rates of convergence of the polynomial's coefficients and also the ability of the polynomial to predict simple waveforms. As a consequence a connection is established between the exponential components of the data and the coefficients of the polynomial that represented it.

A theory is developed to explain this connection and which requires concepts based upon information flow, borrowed from physical optics, which in turn establishes a connection between holograms and information flow. Two papers containing these results were presented at IV Symposium of Biocybernetics, Leipzig [61] and the VIII Congress of Cybernetics in Namur [62].

The proposed theory is investigated thoroughly and for certain generated waveforms an anomaly is found which suggests that the theory is not completely valid, but these variations can easily be included in the proposed theory by extending the coefficients to include complex values. These results were presented at the 3rd Annual Conference 'Recent Topics in Cybernetics' London [63].

ACKNOWLEDGEMENT

The author wishes to thank the members and staff of the Pulton Place Annexe of Chelsea College and the post-graduates in the Electronics Research Laboratory for their worthwhile and lengthy discussions. Grateful thanks are particularly offered to Dr. Fatmi for his assistance, to my colleague Mr. John Ralphs for his advice on presentation, and to Mr. John Parker, also a colleague, for the help given by himself and his wife Teresa in producing a typed copy from my manuscript in such a short time. Needless to say, the author extends sincere thanks to his wife, Susan, for her patience and understanding which, at times, must have been sorely taxed.

The author also wishes to extend thanks to the Science Research Council for funding this project.

INFORMATION FLOW IN CYBERNETIC SYSTEMS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF COMPUTER PROGRAMS (APPENDIX A)

'Step by step, at every step there stands that which is conducive to the next step.'

Professor A.N. Kolmogorov,
Founder of the Axiomatic Theory
of Probability.

CHAPTER ONE

1. INTRODUCTION

1.1 General Information

A first introduction to Cybernetics can be obtained by reading Wiener's book 'Cybernetics : or Control and Communication in the Animal and the Machine'. As each chapter delves deeply in to Wiener's ideas of cybernetics, some help may be required to simplify some of his thoughts. The information content of the book seems immeasurable as, at each fresh reading, more and more information is extracted as patterns emerge and insight is gained into Wiener's understanding of the vast number of topics that fall under the heading of cybernetics.

Kolmorogov's work on the subject of optimum linear filtering produced a polynomial which, in its general form can be adjusted by imposing limits etc. to enable the polynomial to be reduced to any of the more familiar forms of time-series representation.

In his book, Wiener considers time-series and messages and derives a measure of information; the sources he uses, however are of a probabilistic nature. Wiener also shows that the processes which lose information are, as expected, closely analogous to those which gain entropy.

If a distribution of a certain variable is replaced by a distribution of a function of that variable, which has exactly the same values only for different arguments or similarly if in a function of several variables they are all allowed or some of them, to vary unimpeded over their natural range then information is lost. In this precise application of the second law of thermodynamics to communication engineering, Wiener concludes that on average no operation can gain information, and conversely that for an increasingly ambiguous situation, information on average is increasingly gained and not lost.

In 1957, Gabor attempted to build a universal nonlinear filter, simulator and predictor which had one main difference from all previous attempts of a similar nature. This was the use of a learning meathod. In fact, what Gabor built was a highly-adaptable high-speed analogue computer which, in its day, was unique; today, of course Gabor's ideas can easily be simulated on a powerful digital computer, but this does not in any way detract from his great achievement.

The machine which Gabor had constructed made use of Kolmogorov's polynomial; its coefficients were adjusted by a learning process by which their initial values were continually changed in a cyclic sequence in accordance to some predetermined error criterion. Gabor's paper, which was published in 1960 and co-authored by Dr. Wilby and Dr. Woodcock, was to lay the foundation for the work presented in this thesis connected with information flow.

Gabor had visualised how the polynomial could be used as a simulator of other systems. The coefficients can be adjusted for a certain output from a given input rather than for the increase of predictor accuracy. The output and input used was that of another system.

Many papers have been written by Gabor including one entitled 'Light and Information' and another in holography which was read at his Nobel Prize Lecture. Using Gabor's ideas experiments were undertaken which achieved the successful production of holograms. The development of the mathematical relationship between holograms (with their ability to store the total information of a 3D scene) and Kolmogorov's polynomial was subsequently achieved. The link between holograms and a polynomial with capabilities of information extraction was, as a consequence, made.

1.2 Historical

In 1942, Kolmogorov [1] proposed the concept of optimum linear filtering which, later that year, was supported mathematically by Wiener [2]. Wiener then turned his attention to non-linear filters [3, 4] whilst many other people also considered this problem [5, 6, 7, 8]. It was clear that, even if a formal solution could be found, it might not be of practical use. A polynomial (named after Kolmogorov) derived from this filtering theory gives a basic equation which can be used for filtering, predicting and simulating. The work done by Kolmogorov and Wiener was, from a mathematical point of view, very elegant, but in application impracticable because it considered infinite bandwidths and a polynomial of infinite length.

The classic report written by Wiener for the NDRC was nicknamed 'The Yellow Peril' by 'bewildered' engineers because of its colour, and was later published in book form [2]. Bode and Shannon [9] published an article in the proceedings of the IRE in which a successful attempt was prescribed to simplify the derivation of the linear least square smoothing and prediction theory. This described a box with inputs of message m(t) and noise n(t), and with an output of $m(t + \alpha)$ where α is a positive or negative delay. This box thus has the properties of smoothing and prediction.

According to Bode and Shannon, the three main assumptions that the Wiener-Kolmogorov theory makes are

a) that the time series represented by the message m(t) and the noise n(t) are stationary. (A stationary time series is one whose statistics do not change with time. Speech for example, can be considered stationary over short periods of time).

b) the error measure is the mean square discrepancy between the actual and desired outputs, and the box's contents minimise this quantity. The average error is then taken over all possible messages and noise signals, each weighted in accordance with their occurance probability; thus making it probabilistic.

c) the procedure used to smooth and predict is a linear operation (i.e. a linear physically realizable filter).

The theory holds true if all of the assumptions approach perfection, but if any one or more of the assumptions is far from the truth, the associated mathematics becomes very complex and difficult. The Wiener - Kolmogorov theory is of considerable importance in communication theory.

Similarly Shannon's [10] information theory [11, 12] that stemmed from his 1948 paper 'The mathematical Theory of Communication' provided a much needed base for the formulation of a quantitative measure of the commodity dealt with by communication engineers.

1.3 Wienerian Concepts

In 1948, 'Cybernetics' as coined by Wiener [13] was not an attempt to create a new science, but one to unite, within a single discipline, those activities which had until then been associated with different subjects. The word Cybernetics is derived from the Greek κυβερήτης meaning steersman of a merchant roundship.

By bringing together different disciplines under one heading, he hoped to increase the interaction between them which had, until that time, been almost negligible. A suitable example is one in pharmacy and is concerned with the dispersion of drugs in the human body and with the rate of build-up and decrease of the drug level in the blood and bodily wastes, from a single intravenous injection; a problem of this type is reasonably simple in control theory. As a system, the input to the human body is an impulse for which the corresponding output is a response. From this information, a considerable amount can be ascertained about the system, such as its transfer function etc. Similarly, ideas in Communication and Information Theory (eg. the sampling theorem) have important uses in many fields that are concerned with information measuring, extraction etc.

Because Cybernetics is a mixture of disciplines, many definitions have been formulated. Generally speaking, the great names in Cybernetics that followed in Wiener's footsteps (Ashby, George, Pask and many others) have tended to lean Cybernetics in their own particular direction and, thus, its main attribute for each became mathematical or biological or managerial, etc.

On the whole, however, the interpretation used in England is similar to that used in Europe and to some extent Russia. The emphasis being placed in engineering aspects including control and communication, biology and the study of self-organising adaptive systems. This interpretation closely follows Wiener's original ideas.

Wiener classifies time series as a sequence of numerical quantities, distributed in time. A continuous recording of temperature variation, the closing figures of the stock market, or meteorological data are all time series, continuous or discrete, simple or multiple. Until the advent of high-speed digital computers, the only way such slow-changing time series could be analysed was by hand computation or by slide rule. Today more complex and faster-changing time series can be investigated such as telephone signals, television or gunnery data. These studies all belong to the conventional part of statistical theory. Wiener mentions that one of the simplest forms of information measure is the recording of the choice between two equally probable simple alternatives such as a tossed coin giving heads or tails. What is the amount of information, of an accurately measured value, lying between two limits? Given that this quantity can be specified by a binary number, it then needs to be infinitely long in order to define the number accurately, the number of choices made and the amount of information as a consequence is infinite. Thus the more decisions there are the more information we have. However, such accuracies in measuring a number are not possible in a practical situation.

density

Given f(x) as a probability function, then the total area under this curve must be unity. Consequently the average logarithm of the breadth of the region under f(x) is an average of the height of the logarithm of the reciprocal of f(x). Therefore a reasonable measure of the amount of information associated with the curve f(x) is

$$\int_{-\infty}^{\infty} \left[\log_2 f(x)\right] f(x)dx \qquad 1.3.1$$

This, as defined by Wiener, is the negative of the quantity usually defined as entropy. Wiener goes on to assert that information from independent sources is additive; he also states that no operation on a message can gain information. This is an obvious statement when considered and has similarities to the conservation of energy law. A message which contains a certain amount of information can be made to lose some of its content or, by manipulation, yield its contents more readily, but under no circumstances can the nessage be made to yield more information than it holds by definition.

1.4 Gaborian Concepts

In Gabor's 'Light and Information' published in 1951 [14] the author points out that light is our most powerful source of information in the physical world. Aldous Huxley remarked that our civilisation owes its existence largely to the fact that vision is an objective sense. Certain animals have highly developed senses of smell and hearing, but no matter how highly developed these senses are, the animals can never develop science as we know it as they lack sight developed to the same extent.

Gabor's view of information theory can be approached in two steps. The first step is to specify the degrees of freedom of the phenomenon, such that the degrees are always discrete and their number finite. This according to MacKay specifies the structural aspect of information.

Once the coordinates are decided upon, the second step is to associate a measure with each coordinate. However, as some error will always be present in the measurement, a probability that the measurement is within certain limits has to be given.

In practice the coordinates depend upon the system as illustrated in Eddington's [15] parable of the fishing net - if a fisherman uses a net that has holes two inches across, he cannot expect to catch fish smaller than two inches. Similarly, given a perfect bandpass filter with a bandwidth spreading from 100 Hz to 1 kHz, then any incoming signal, after passing through the filter, cannot be expected to contain any frequency components above and below the filter limits. This has an analogy with the sampling theorem in that to extract the information from a given

signal it must first be assumed that the waveform contains the information and second that the selecting or sampling procedure used is sufficient to extract the information. The sampling procedure can be considered as a transformation from one domain to another such that the waveform is transformed and thus rearranged so that the information is more easily available. Transformations of this type can lose information by inefficient mapping.

An inefficient mapping is one that reduces the original waveform in size, eg. a waveform of n points can, after transformation, be represented by m values where $m < n$, (n being the minimum number of points in the original domain that represents the original signal). The transformation of a system represented by n elements is performed by mapping each element from the original domain to the transformation domain. If the original signal contains redundancies and requires reducing, the mapping is not one to one, and one point in the transformation domain has been mapped from more than one point from the previous domain. Similarly, redundancy can be added where a point in the original domain is mapped into more than one point in the final domain.

Usually the original signal contains a certain amount of redundancy which can be removed by the transformation to give a more efficient representation of the signal. For example, speech, according to Shannon [10], contains a maximum of 80% redundancy in a perfect noise-free environment; thus, for efficient signal processing, it is best to remove the natural redundancy and add electronic redundancy in the form of error correction codes to give the maximum usage of a given bandwidth.

Gabor's paper 'Theory of Communication' [16] presented a unit of information called a logon, the idea being related to Nyquist's [17] work on telegraph signal speed; a very important and intriguing aspect of this work was that, due to its non-statistical nature, it did not rely on probability. Unfortunately the concept did not prove popular!

Gabor's ideas that later lead to holography, [18] (the three dimensional representation of objects) are also very intriguing. The idea of total informational storage of a three dimensional scene is an example of a transformation where redundancy is added; this is easily shown since a 3-dimensional scene can be constructed from a small portion of the original hologram with only marginal increase in background noise. Similarly, by taking the hologram and shining a laser directly onto a point, a projection of a particular view is obtained. When the laser is scanned, different views are shown; this relates to a flow of information between adjacent points on the hologram. By relating this information flow to Kolmorgorov's polynomial and, consequently, the polynomial to total informational storage, the polynomial is capable not only of adequately representing a waveform but can also be used to monitor the flow of information that takes place.

In 1960, Gabor [19] and two colleagues attempted to build a universal non-linear filter predictor and simulator which optimized itself by a learning process. The machine was based on Kolmorgorov's polynomial and the product of the work done by Kolmogorov and Wiener in 1942 as mentioned earlier. The maximum size of polynomial was 94 terms comprising 18 linear terms and those of higher order. Each term had associated with it a coefficient or weighting factor that could be adjusted in accordance with an error criterion.

Given a Stochastic process f(t) with limited frequency band F, it is necessary to sample the waveform at the Nyquist rate thus giving

$$f(t) = \sum_{n=-\infty}^{\infty} f(t - n\Delta) \qquad 1.4.1$$

This forms the input data. For the construction of the polynomial, a consecutive set of n data points are used $f(t_1)$, $f(t_2)$.....$f(t_n)$. Having formed the polynomial from the n data points, it is made equal to $f(t_{n+1})$ which is the next consecutive data point in the original set. The polynomial in question is of the following form

$$f(t) = \sum_{i=}^{n} a_i f(t - i\Delta) + \sum_{i}^{n} \sum_{j}^{n} a_{ij} f(t - i\Delta) f(t - j\Delta) + \sum\sum\sum..... \qquad 1.4.2$$

and can be split into groups. The linear group consists of the linear terms which comprises the data points each with its own coefficient. The quadratic group containing quadratic terms is constructed by combining any two of the n data points, again each term having a coefficient associated with it, and so on.

The coefficients need to be adjusted such that the polynomial is equal to the n + 1 th data point; any difference is regarded as an error. The polynomial is constructed from a set of n points which are obtained by placing a window of length n over the total sampled history to allow a maximum of n linear terms, n(n + 1)/2 quadratic terms, n(n + 1)(n + 2)/6 cubic terms etc. Thus it can be seen that the size of the polynomial rapidly increases with the value of n.

Gabor's machine, with its facilities for a maximum of 94 terms, of which a maximum of 18 can be linear, suffers restrictions with respect to how many higher order terms can be utilised. For example if n is 12 (i.e. 12 linear terms) then the maximum number of quadratic

terms is 78 (i.e. n(n + 1)/2) ignoring all higher order terms. Therefore in this example no more linear terms can be employed without disregarding the corresponding quadratic terms.

Once the machine was built, it was tested with some simple data sets a) phase shifting and scaling of a sinewave and b) filtering a sinusoidal signal with added noise. The sets comprised two coefficients for the first test and what appears to be a random choice of six coefficients for the noisy sinewave in the second test. One of the main problems of the Kolmogorov - Wiener filter theory is that it relates to systems with infinite bandwidth because, in practise, the bandwidth of a system is always finite. In a frequency band F and a time interval T there are 2FT degrees of freedom. Therefore any band-limited signal in a finite time can be represented by a finite number of parameters. These parameters or samples can be thought of as a series of (sin x)/x pulses on the samples and of a proportional height. In fact the waveform f(t) is then said to be adequately represented by its samples. This, of course, is Shannon's [10] well known sampling theorem:

$$f(t) = \sum_{t_n=-\infty}^{\infty} f(t_n) \frac{\text{Sin } 2\pi f(t - t_n)}{2\pi f(t - t_n)} \qquad 1.4.3$$

The data was recorded on a magnetic tape using pulse rate modulation and the required delays are achieved using staggered heads. On one track of the tape is stored the data which acts as a target function; if the machine is used as a predictor, this data is exactly the same as the other 18 tracks, and interpolation or extrapolation is achieved by setting the corresponding delays of the recorder's playback heads. Similarly, if used as a simulator, the target function becomes the output of the system that is to be simulated, and the data on the other tracks represent the system input. The coefficients are adjusted such that the value of the

polynomial approaches the target function, the difference being used to calculate the mean square error; further coefficient adjustment is used until this error is minimised. The criterion used was as suggested by Wiener and Kolmogorov.

Different stratagies were employed by Gabor and included Southwell's relaxation method for always adjusting the coefficient that gives the greatest reduction in the error; unfortunately, three main problems exist which outweigh any advantages:

1) greater time consumption
2) greater storage
3) scanning the coefficients to find which gives the greatest reduction in error.

An alternative strategy he used was to adjust each coefficient in turn because it took 100 seconds to do the relevant calculation for each coefficient which itself takes only 1 second to adjust: this method obviously converges quicker than Southwell's method. A slight modification suggested by Gabor was that any coefficients that made little or no difference to the error could be ignored for the next N training runs (a training run is used to describe the situation when each coefficient in the polynomial has been adjusted once). This gives a compromise between the two strategies already mentioned.

Gabor suggests that for an m coefficients, $\frac{1}{2}m^2$ training runs will be sufficient to reduce the mean square error to a reasonable level. A plot of the mean square error against the number of training runs rapidly decreases, but even for only two coefficients and a simple noise free sinewave as the input data, the error was still greater than 25% of the initial error. These error plots used by Gabor take the initial error as 100% and subsequent errors are

relative to this on a percentage basis. Chapter two clearly shows that the initial error can be many orders of magnitude above an acceptable error level.

Young [20] wrote an Algol programme that simulated Gabor's machine and which was later translated into Fortran by Muftogulu [21] for work on hydrological systems. As the basic equation was understood, the learning method used by Gabor was now open to investigation. A modified version of Muftogulu's programme was planned and later a simpler version of the programme was written to enable the links between the coefficients and the information of the time series to be investigated.

1.5 Cybernetics and Prediction

Any system that can be represented by a black box with an input and an output can be considered as a cybernetic system, and these simplified block diagrams showing how the system functions enable a system to be more readily understood. Complex systems can be reduced in complexity by this method and as a consequence the more one understands them, the more one can adjust and simplify the actual operation of such systems. For example the economics of a country, a highly non-stationary and stochastic system, or a large industrial complex such as a petrochemical works can, with the aid of simplified block diagrams, be easily interpreted; once this understanding develops, the actual system can be simulated or modified, the end result being to maximise efficiency.

Every cybernetic system must have an input and output that can be represented by a discrete or continuous time series. Obviously, if a system has more than one input and/or outputs, the first and last block of the diagram then contain whatever elements are necessary to combine or split the inputs and outputs. If any of the inputs do not interact and thus produce a separate output or outputs, they can then be separated from the overall system and the original system be therefore simplified. Similarly, any smaller internal system that is complete in itself can be represented with the whole, but separately.

The input thus contains information in one form or another which is transformed by the system into the output. The transformation that takes place is one of extraction - of extracting information from the input, given certain initial informational conditions that are held

within the actual system. The information itself flows within the system, entering as an input and leaving as an output.

Prediction is a time domain analysis, as compared with Fourier analysis which takes place in the frequency domain, and has the obvious advantages that any data can be analysed directly with no requirement for initial transformation. Time series analysis developed in the fields of statistics, economics and communication and, is the main method used for the analysis of dynamic systems. Recently, however, control systems theory, using state space concepts and time domain analysis, has made great advances in the analysis of dynamic systems.

Kolmogorov's investigation into linear extrapolation of stationary random processes used probabilistic theory. Krein [22] and Yaglom [23, 24] similarly both used probabilistic concepts, and further work was undertaken by Brown [29] on the linear prediction of band-limited [25,26,27,28] processes, and by Hajek [30] on the prediction of stationary processes with convex correlation functions.

The following references relate to the wide usage of the mean square error criterion. Davisson [31] uses a steady state mean square error of an adaptive linear estimator as applied to stationary data. Multi-dimensional least-square adjustment was considered by Grafarend and Kelm [32] for point and interval linear estimations using tensor algebra and statistical methods. A similar method, the integral squared error is used by Garudachar [33] in finding the optimal technique for the linearisation of second-order non-linear differential equations. Learning algorithms, and their convergence rates, and the efficiency of the learning process were studied by Gulyas [34] and Ainsworth [35] respectively.

Prediction can be of use in a variety of subjects to provide a fresh approach to existing problems. One such difficulty exists in neurophysics where work at present is aimed at trying to establish that EEG signals as picked up on the scalp of a patient actually carry information about the state of the brain. The signals, it is believed, contains the summation of different internal rhythm's the common ones being alpha, beta and delta, whose relative heights give information about the medical status of the body and brain. Papers by Bohlin [36], Fenwick [37], Gersch [38] and Wennberg [39] discuss modelling and simulation of EEG signals.

Seismic vibrations, earthquakes, oil exploration and general geological surveys are all examples where time domain analysis has been used or could be used. Papers in Geophysics by Robinson [40] and Treital [41] deal with the use of predictive decomposition in a model; which represents a section of a seismic trace as the convolution of a random spike train with a minimum delay waveform. A similar problem arises in the analysis of voiced speech except the random spike train is replaced by a quasiperiodic [42] impulse train.

There have been various attempts in communication systems to compress bandwidth : Robinson and Cherry [43] used optimum encoding; Elias [44] coined the term predictive coding in 1950, and this term has been used by Atal and Schroeder [45] for speech signals; Sciulli and Campanella [46] used it in connection with multi-channel telephony and by Kobayashi and Bahl [47] in connection with image data compression; Davisson used straight line interpolation [48] and prediction [49] in data compression and attempted to formulate a relevant theory [50].

Spectral analysis techniques [51] have evolved with the advent of high-speed computers. In terms of optimality, the more common transform techniques can be ordered thus : Fourier, Hadamard [52] and Karhunen-Loeve [53]. Hadamard transforms have been used mainly with image coding, but faster Fourier transforms have been formulated [54, 55, 56] for use with digital computers and discrete time series. Speech analysis and synthesis [57, 58] and prediction [59,60] have been major contributors to the advancement of the work on linear prediction. The spectrum of speech contains four major peaks or formants which are the result of resonances in the vocal tract. Linear predictions of speech can be accomplished with 12 to 14 coefficients, this number being arrived at by a trial and error process involving the use of a variety of coefficients from 1 to 20, and then selecting the number of coefficients that produce an acceptable error. A relationship between the number of coefficients, the corresponding error and the waveform's spectrum was found and investigated.

CHAPTER TWO

2 EXPERIMENTAL RESULTS

2.1 Experimental studies based upon a Learning Program

In October 1971 an investigation was started into the possibility of linking Cybernetics with Communications, using a program developed earlier by Fatmi, Young and Muftogulu. Substantial modifications were required to adapt the program to this application, and the modified version is shown in Appendix A program A1. The results obtained are given in tables 2.1.1 to 2.1.5.

The information flow in an elementary cybernetic system is shown in figure 2.1.1. It consists of an input and an output, with information flow between system A and system B through a two-way connection. System A comprises a simulator which attempts to model itself upon system B. The outputs of the simulator and system B are compared, their difference being an error which is used via a self-optimising element to adjust the simulation in such a manner as to reduce itself.

If system B is a simple delay system, A becomes a predictor. A learning process is used to adjust the simulator or predictor, and which takes a finite time to reach an optimum solution. The predictor simulator is in the form of a polynomial of n adjustable parameters (coefficients). Obviously the value of n depends upon the complexity of system B. In his paper entitled 'A universal non-linear filter,

predictor and simulator which optimizes itself by a learning process, Gabor suggests that if there are m variable parameters, they will need to be adjusted approximately $\frac{1}{2}m^2$ times if they are to approach their optimum values. Therefore the validity of this approximation was investigated together with any other relationship that might exist between the number of training runs (tr), mean square error (mse) and the number of parameters (tp).

The data used was a deterministic Ramp function and each calculation was stopped when the forecast value was within 10% and then 1% of the correct value. The results are shown in tables 2.1.1 and 2.1.2.

To test for any relationships that might exist between any two of the three variables tr, mse and tp, the correlation coefficient was calculated between two variables with the third held constant. For a possible relationship between, for example, training runs and mean square error, the correlation coefficient would be expressed as cor(tr,mse).

In table 2.1.1, cor(tr,mse) = -0.7348. This suggests an inverse relationship which is acceptable as the more training there is, the smaller the error tends to be. However, cor(tr,tp) = -0.5234 suggests that the more terms there are in the polynomial, the less training there needs to be, it thus converges more rapidly. Table 2.1.2 gives similar results for cor(tr,mse) = -0.7184, but cor(tr,tp) = -0.0042 implies no relationship between tr and tp.

The above criteria for stopping the training was considered not to be very accurate because, although the parameters obtained might

give a good prediction in this case for 11, they would not be suitable for predicting, say 12. This is also shown by the wide variety of the mean square error in tables 2.1.1 and 2.1.2. Thus, the criteria for stopping each training was for the mean square error to have been minimised or to be less than 0.1, 0.01 and 0.001 as shown in tables 2.1.3 to 2.1.5.

The degree of interaction, n, has values of one, two and three, and relates to the complexity of the polynomial: the n = 1 polynomial contains linear terms only; the n = 2 polynomial contains linear and quadratic terms only; the n = 3 polynomial contains linear, quadratic and cubic terms. For a degree of interaction 1, the training runs are between 0 to 200 for mse less than 0.1, 0 to 400 for mse less than 0.01, and 100 to 600 for mse less than 0.001. The respective maximum differences between the forecasts and 11 are 5.2%, 1.6% and 0.54% and are acceptable accuracies. For a degree of interaction 2 the training runs from 200 to 400, 700 to 1,200 and 1,100 to 2,100 for mse less than 0.1, 0.01 and 0.001 respectively, and their maximum differences from 11 are -10.96%, 3.56% and -1.09% respectively. This shows that for n = 2, the percentage differences are doubled and that the predicted answer approaches 11 from above rather than from below when n = 1 is used. For n = 3, the training runs range from 100 to 200, and 900 to 2,400, and the errors are -5.3%, 8.5% for mse less than 0.1, 0.01. Thus, for n = 3, the predicted answer approaches 11 from below and then above. Thus the best system for a ramp input seems to be one with a linear degree of interaction. Cor(tr,tp) = -0.0587, 0.3155 and 0.4498 for the three descending values of mse imply that there is a possible relationship forming, but it does not appear to be of the form $\frac{1}{2}m^2$.

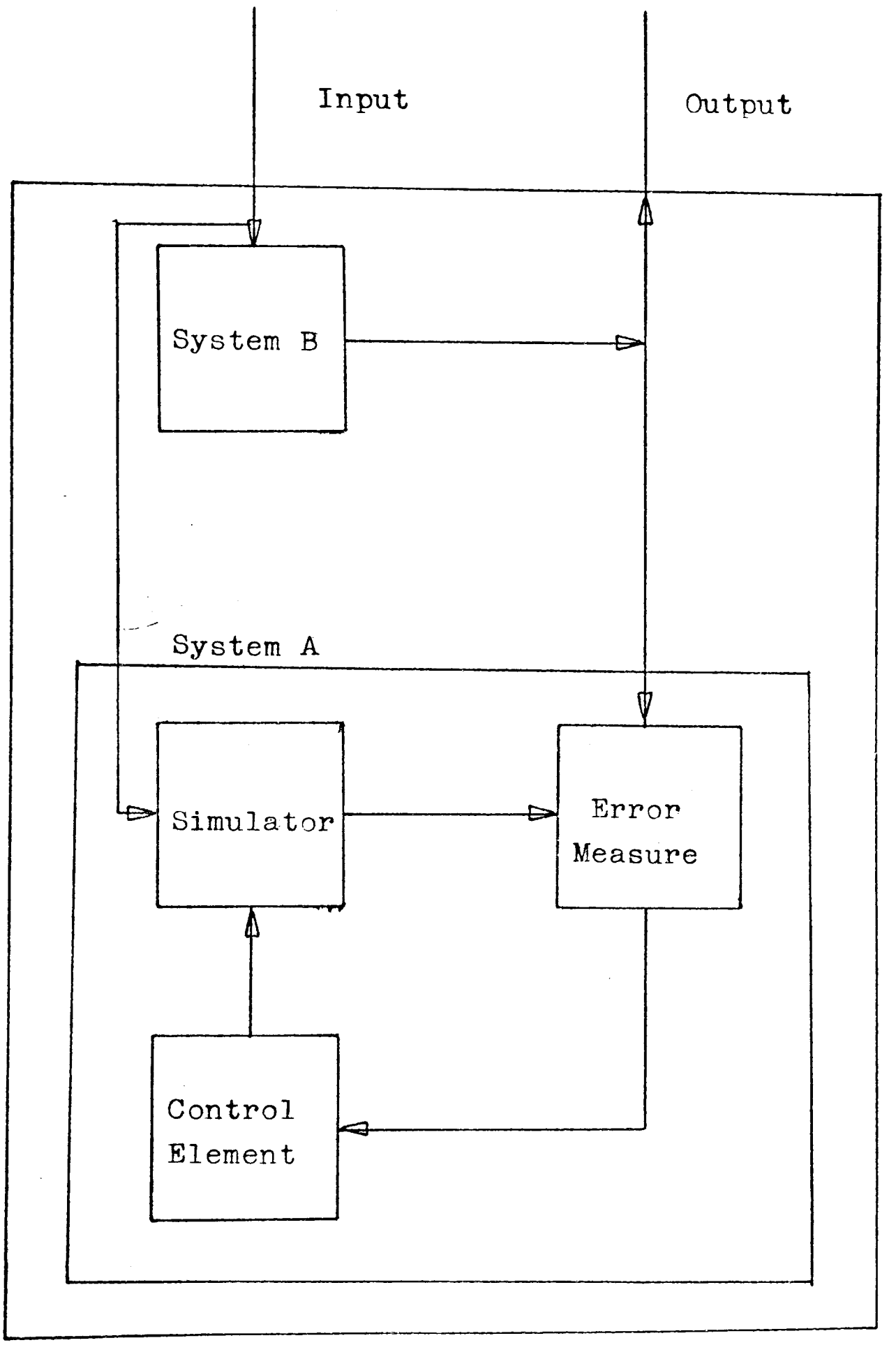


Figure 2.1.1 Elementary Cybernetic System.

J Size of sub-data group.
N Degree of interaction.

IU(1) Number of linear terms in the polynomial.
IU(2) Number of quadratic terms in the polynomial.
IU(3) Number of cubic terms in the polynomial.

IT Total number of terms in the polynomial.
ITA Number of training runs.
EEA Mean square error.

RESULTS.

J	N	IU(1)	IU(2)	IU(3)	IT	ITA	EEA	FORECAST VALUE :
2	1	2	0	0	2	29	.5945957	12.098
3	1	3	0	0	3	44	.4800038	12.094
4	1	4	0	0	4	39	.4031998	12.096
5	1	5	0	0	5	35	.3376187	12.093
2	2	2	3	0	5	1	.7546047	11.798
3	2	3	6	0	9	3	1.1910828	11.879
4	2	4	10	0	14	4	1.4927501	11.885
5	2	5	15	0	20	4	1.7985543	12.037
2	3	2	3	4	9	1	.7363222	11.493
3	3	3	6	10	19	1	1.4901426	11.804
4	3	4	10	20	34	2	1.7707466	11.439
5	3	5	15	35	55	2	2.1730570	11.357

Correlation coefficient between training runs and mean square error. COR(tr..mse) -.7348
Correlation coefficientbetween training runs and terms in the polynomial. COR(tr..tp) -.5234

Table 2.1.1 Training stopped when forecast reached plus or minus 10% of 11.

J Size of sub-data group.
N Degree of interaction.

IU(1) Number of linear terms in the polynomial.
IU(2) Number of quadratic terms in the polynomial.
IU(3) Number of cubic terms in the polynomial.

IT Total number of terms in the polynomial.
ITA Number of training runs.
EEA Mean square error.

RESULTS.

J	N	IU(1)	IU(2)	IU(3)	IT	ITA	EEA	FORECAST VALUE :
2	1	2	0	0	2	425	.0059376	11.110
3	1	3	0	0	3	195	.0047427	11.109
4	1	4	0	0	4	126	.0039528	11.109
5	1	5	0	0	5	96	.0032752	11.108
2	2	2	3	0	5	8	.3968914	11.055
3	2	3	6	0	9	8	.6683277	11.002
4	2	4	10	0	14	7	.9454828	11.076
5	2	5	15	0	20	7	1.0039487	10.900
2	3	2	3	4	9	3	.5384823	11.098
3	3	3	6	10	19	3	1.0015182	11.044
4	3	4	10	20	34	177	.0560771	10.894
5	3	5	15	35	55	169	.0529944	10.891

Correlation coefficient between training runs and mean square error. COR(tr..mse) -.7184
Correlation coefficient between training runs and terms in the polynomial. COR(tr..tp) -.0042

Table 2.1.2 Training stopped when forecast reached plus or minus 1% of 11.

J Size of sub-data group.
N Degree of interaction.

IU(1) Number of linear terms in the polynomial.
IU(2) Number of quadratic terms in the polynomial.
IU(3) Number of cubic terms in the polynomial.

IT Total number of terms in the polynomial.
ITA Number of training runs.
EEA Mean square error.

RESULTS.

J	N	IU(1)	IU(2)	IU(3)	IT	ITA	EEA	FORECAST VALUE :
2	1	2	0	0	2	183	.0991303	11.448
3	1	3	0	0	3	96	.0978827	11.494
4	1	4	0	0	4	66	.0959750	11.535
5	1	5	0	0	5	52	.0927630	11.573
2	2	2	3	0	5	313	.0999985	10.240
3	2	3	6	0	9	287	.0998259	10.096
4	2	4	10	0	14	278	.0997618	9.961
5	2	5	15	0	20	279	.0996947	9.794
2	3	2	3	4	9	149	.0998343	10.641
3	3	3	6	10	19	145	.0997162	10.582
4	3	4	10	20	34	135	.0999497	10.508
5	3	5	15	35	55	129	.0999461	10.402

Correlation coefficient between training runs and mean square error. COR(tr..mse) .6493
Correlation coefficient between training runs and terms in the polynomial. COR(tr..tp) -.0587
Correlation coefficient between mean square error and forecast value. COR(mse..fv) -.7216

Table 2.1.3 Trainingstopped when mean square error was less than 0.1

J Size of sub-data group.
N Degree of interaction.

IU(1) Number of linear terms in the polynomial.
IU(2) Number of quadratic terms in the polynomial.
IU(3) Number of cubic terms in the polynoial.

IT Total number of terms in the polynomial.
ITA Number of training runs.
EEA Mear square error.

RESULTS.

J	N	IU(1)	IU(2)	IU(3)	IT	ITA	EEA	FORECAST VALUE :
2	1	2	0	0	2	381	.0099061	11.142
3	1	3	0	0	3	171	.0098795	11.157
4	1	4	0	0	4	109	.0097588	11.171
5	1	5	0	0	5	82	.0094904	11.183
2	2	2	3	0	5	1186	.0099880	10.760
3	2	3	6	0	9	782	.0099963	10.714
4	2	4	10	0	14	700	.0099482	10.672
5	2	5	15	0	20	700	.0099880	10.618
2	3	2	3	4	9	2328	.0099968	11.479
3	3	3	6	10	19	1665	.0099994	11.582
4	3	4	10	20	34	1540	.0099982	11.710
5	3	5	15	35	55	944	.0099995	11.936

Correlation coefficient between training runs and mean square error. COR(tr..mse) .6308
Correlation coefficient between training runs and terms in the polynomial. COR(tr..tp) .3155
Correlation coefficient between mean square error and forecast value. COR(mse..fv) .0564

Table 2.1.4 Training stopped when mean square error was less than 0.01.

J Size of sub-data group.
N Degree of interaction.

IU(1) Number of linear terms in the polynomial.
IU(2) Number of quadratic terms in the polynomial.
IU(3) Number of cubic terms in the polynomial.

IT Total number of terms in the polynomial.
ITA Number of training runs.
EEA Mean square error.

RESULTS.

J	N	IU(1)	IU(2)	IU(3)	IT	ITA	EEA	FORECAST VALUE :
2	1	2	0	0	2	579	.0009899	11.045
3	1	3	0	0	3	246	.0009972	11.050
4	1	4	0	0	4	152	.0009923	11.054
5	1	5	0	0	5	112	.0009709	11.059
2	2	2	3	0	5	2059	.0009976	10.924
3	2	3	6	0	9	1278	.0009964	10.910
4	2	4	10	0	14	1121	.0009975	10.896
5	2	5	15	0	20	1122	.0009952	10.880
2	3	2	3	4	9	9048	.0009998	11.152
3	3	3	6	10	19	9045	.0009999	11.184
4	3	4	10	20	34	----	-------	------
5	3	5	15	35	55	----	-------	------

Correlation coefficient between training runs and the mean square error. COR(tr..mse) .4615
Correlation coefficient between training runs and terms in the polynonial. COR(tr..tp) .4498
Correlation coefficient between mean square error and forecast value. COR(mse..fv) -.0551

Table 2.1.5 Training stopped when mean square error was less than 0.001.

2.2 Experimental studies based upon concepts of Physical Optics

As the control of the Information Flow was not effective, another experiment was continued to overcome these limitations. One of the main problems was that the first program included linear, quadratic and cubic terms - and, at this time no method was known to select the correct number of terms, and certainly no method was known to determine what order of polynomial was desired. A new program was developed which was restricted to linear terms only, and which is given in Appendix A, program A2. In this experiment, investigations were carried out on the information flow in a cybernetic system using concepts borrowed from physical optics. Figure 2.2.1a shows the general arrangement. The experimental results are shown in tables 2.2.1 to 2.2.4.

Each point on a hologram contains the total information with respect to a particular view. The equation which represents a point on the hologram is remarkably similar to the polynomial used by the predictor; this suggests that the polynomial used can store the total information. Thus the simpler the cybernetic system, the smaller the information flow tends to be.

The first experiment was to examine holograms, the first and easiest approach actually being to make one. The laser used was a Helium Neon, approximately two milliwatts, together with two mirrors, a beam splitter, a stand for the object and a holder for the photographic film, plus chemicals, etc. for developing the hologram.

The objects used were a small model of a red Mini car, a horse and gate, and a pair of bolts. The equipment was arranged as shown in figure 2.2.1b and the objects placed in turn on the stand.

Before taking a hologram it is necessary to check the following points 1) a ratio of approximately 3:1 must be maintained between reference and reflected beams. 2) Stray light from outside or from the laser must be eliminated - this applies especially to secondary reflections. 3) Correct temperatures of chemicals in developing trays must be maintained.

All the lights were extinguished and the film was taken out of its light-proof wrapping and placed in the film holder, making sure that the remaining films were securely wrapped in the light-proof wrapping and the box closed. Various exposure times were used within the range 4 to 10 seconds, the exposure being achieved by switching the laser on for a predetermined time. While the exposure was being made, any movement was kept to a minimum as a relative motion of even a few millionths of an inch between the target and other components can destroy the image. After the exposure, still in absolute darkness, the film was placed in developer for 6 minutes, then in a stop bath to remove any developer, and finally into a fixing tray. After a relatively short time, the lights were switched on at this stage.

To view the hologram it was placed back into its holder and with the laser switched on, an image formed in exactly the same spot as the original object

A hologram has many properties, one in particular being, the ability to show a single view. If the laser is scanned across the hologram, many separate views can be shown. Therefore each point contains the total information of one particular view. Obviously, the polynomial equation of a point on a hologram is complex due to the complexity of the system and the high amount of information flow.

By restricting the polynomial to simple systems, the number of terms can be reduced. Such a reduction could involve only linear terms; this would also simplify the programming. The operational method of the program is explained in the chapter on theory, the principle being that, for a given input and a number of coefficients, the program cyclically adjusts the coefficients until some error measure is minimised. The coefficients are the weights of a weighted linear summation of the input and the adjustments minimise the mean square error between the polynomial and a desired output, as shown in the flow diagram, figure 2.2.2.

The experiments showed that a sinewave or a sinh function needed two linear coefficients, C_1 and C_2, to completely represent them, and in both cases C_2 was minus one the corresponding value of C_1 depended on the sampling distance. See table 2.2.1. It was also noted that a ramp needed only two linear coefficients; C_2 was again minus one and C_1 equal to two. This is the minimum number of coefficients required to represent the above function, although obviously a greater number than this can be used.

Program A2 was designed for only two linear coefficients and also generated its own input data. For an input of 10 sin x and an output of 100 sin x and for the sinewave sampled at 30^{o} intervals, the C_1 and C_2 coefficients are -1.0 and 1.73205 respectively; for an input scaled by 10, the coefficients are similarly scaled at -10.0 and 17.3205, and for input scaling of 1.1, the coefficients are -1.1 and 1.90526. Detailed results are given in table 2.2.2. and 2.2.3

If the graph of the number of training run against mean square error and log mean square error (figure 2.2.3) are compared, it is evident that after approximately 60 training runs, (i.e. each coefficient has been adjusted sixty times) the mean square error has dropped to an acceptable level of accuracy of less than 10^{-11}. Table 2.2.3 shows the results of using input data (mainly 10 sin x + 1) that require more than two linear coefficients. The results show that after 25 training runs, the mean square error reached a minimum value; log mean square error, with similar characteristics, is given in graph figure 2.2.3.

Tables 2.2.4 and 2.2.5 show that if the data is scaled by 10 or 100, the mean square error for the same amount of training is 100 or 100,000 times larger respectively, although the coefficients remain the same. Also, if a sinewave is sampled at closer and closer intervals, it approaches a ramp, a result which is to be expected.

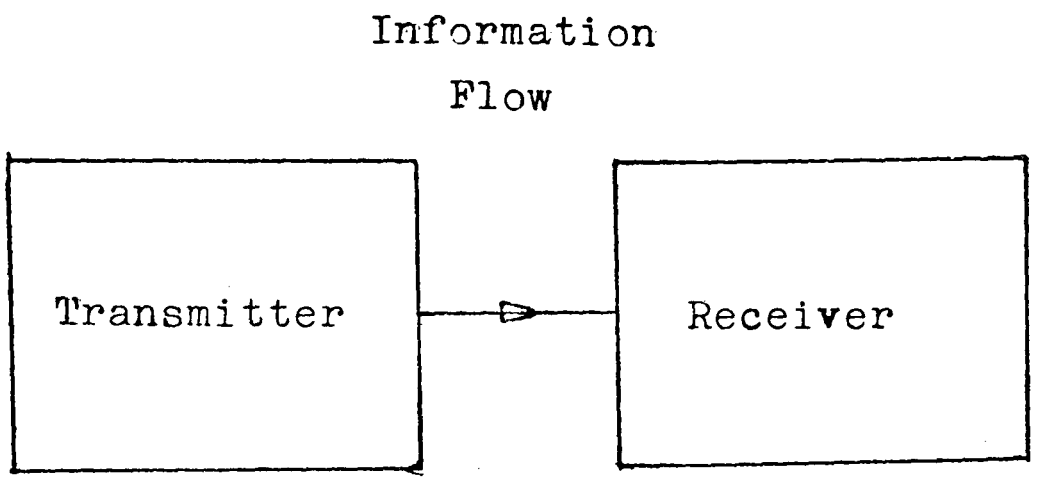


Figure 2.2.1a Perfect Communication System.

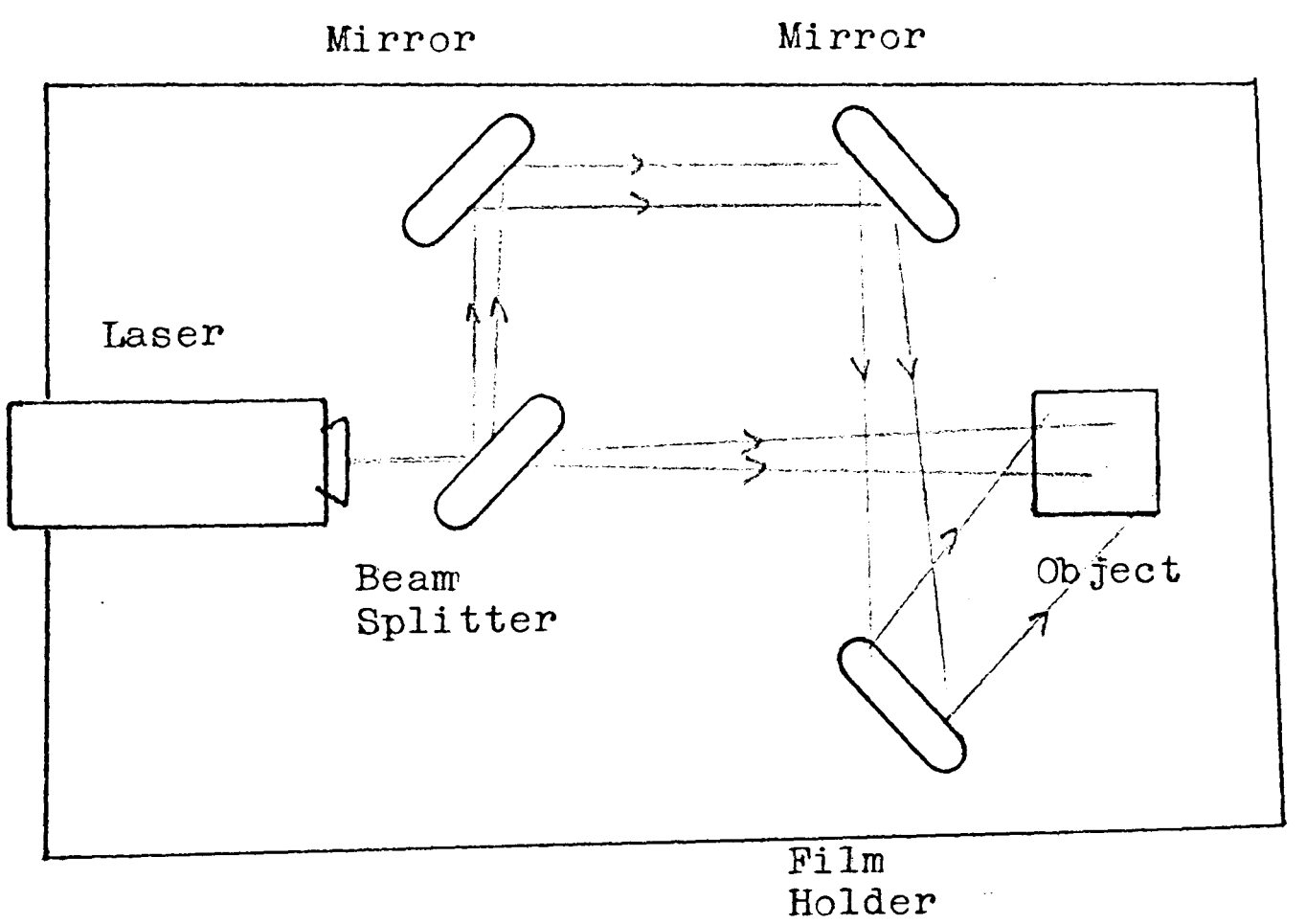


Figure 2.2.1b Taking a Hologram.

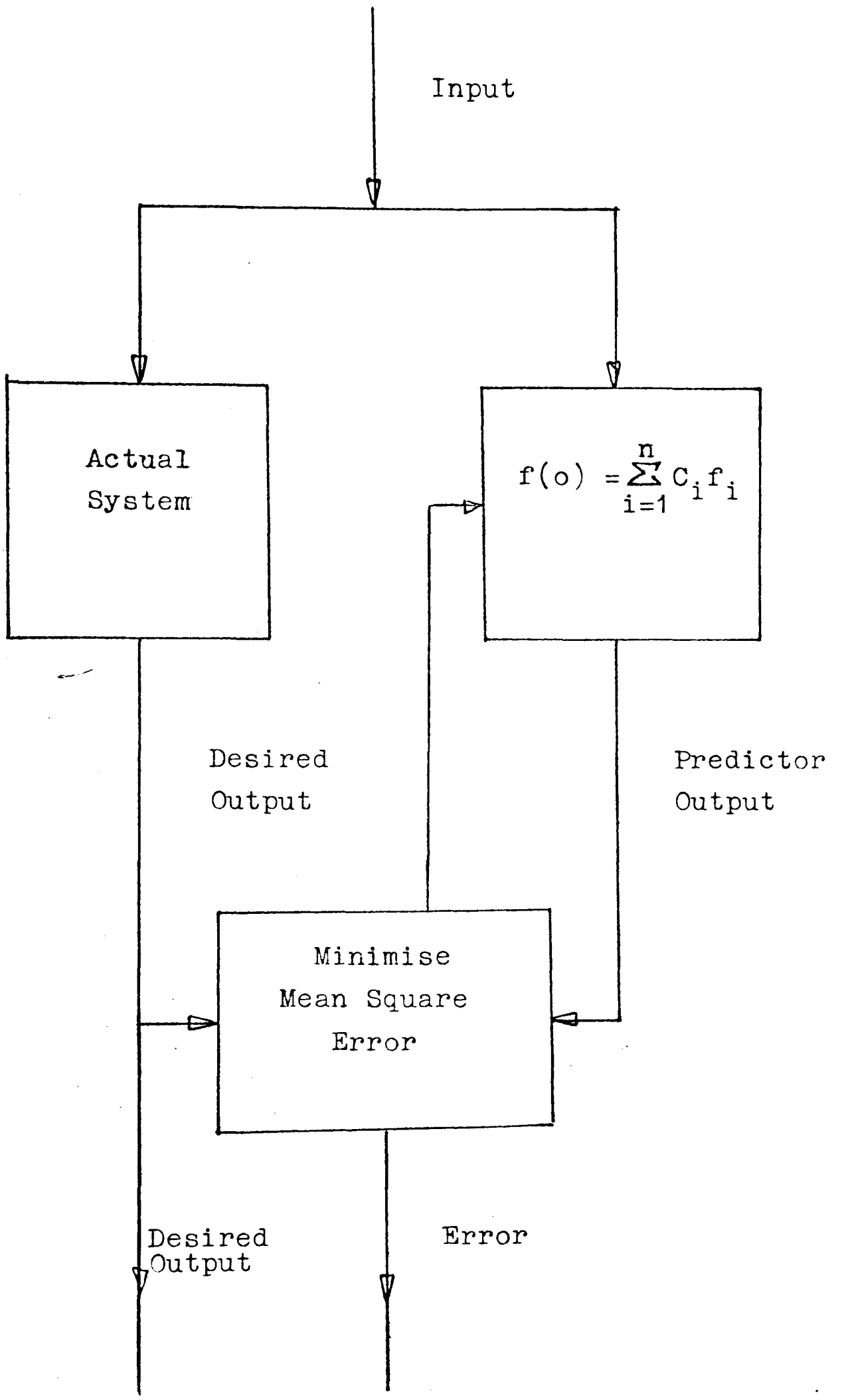
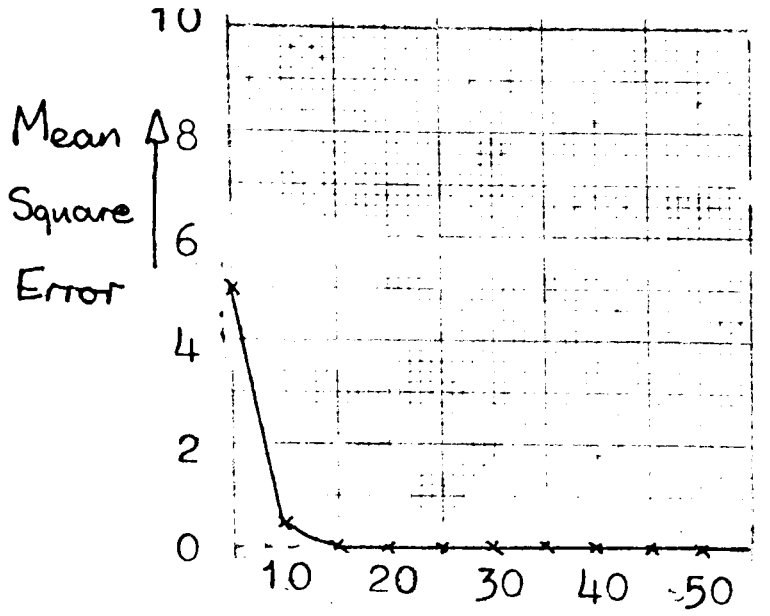
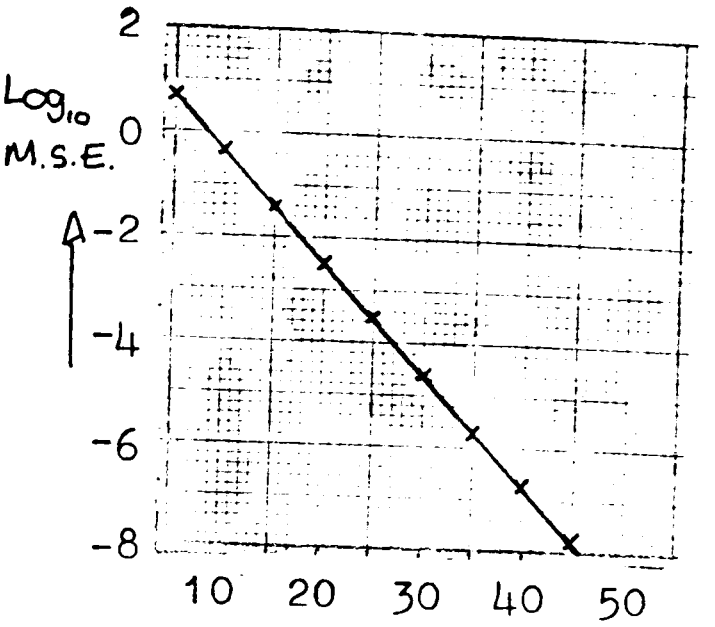
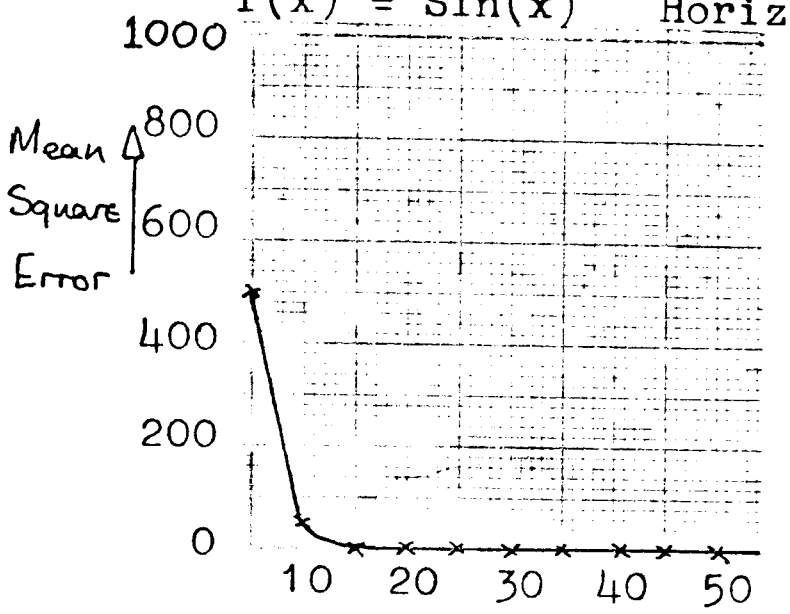


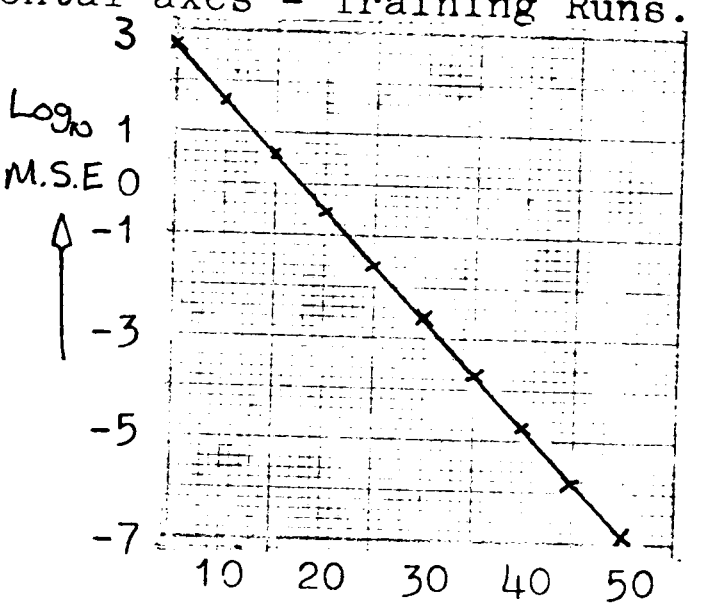
Figure 2.2.2 Simplified Cybernetic System.



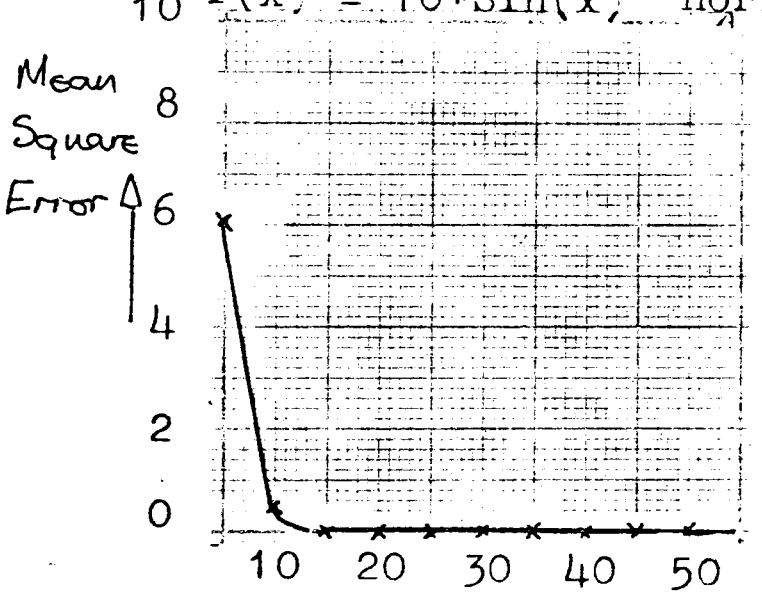


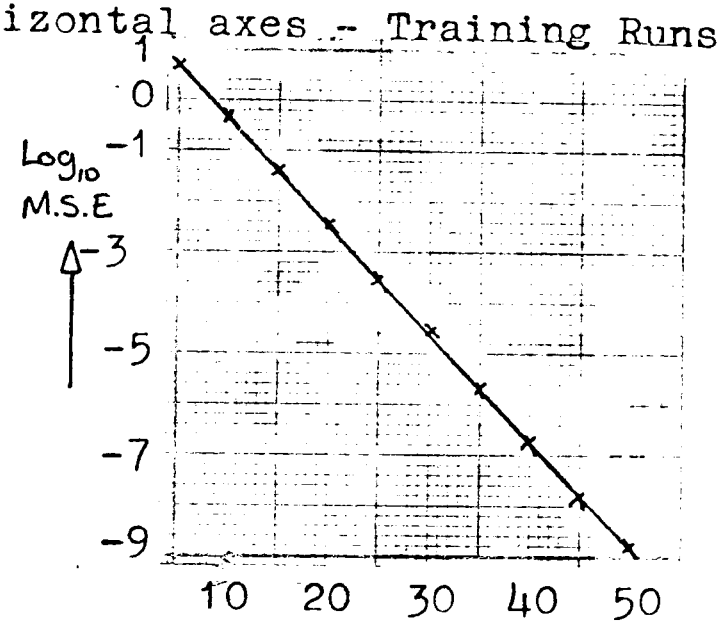
f(x) = Sin(x) Horizontal axes - Training Runs.



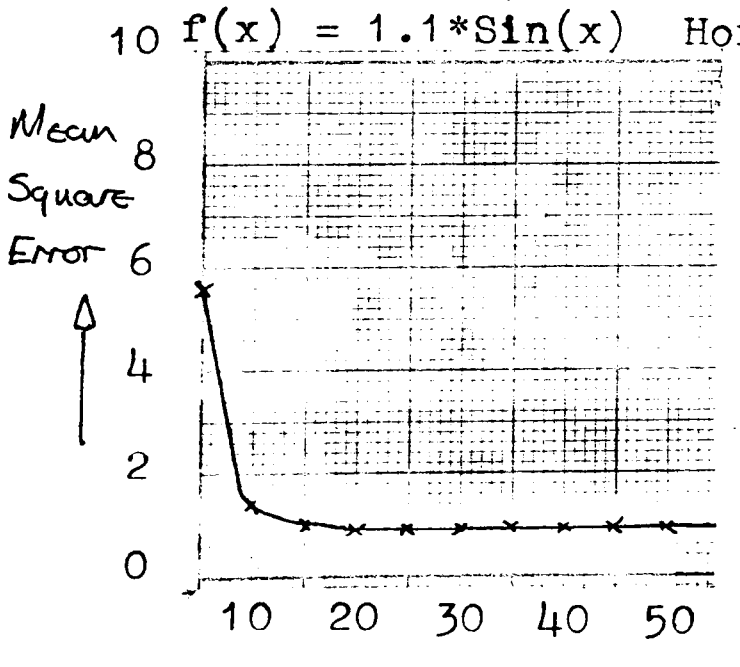


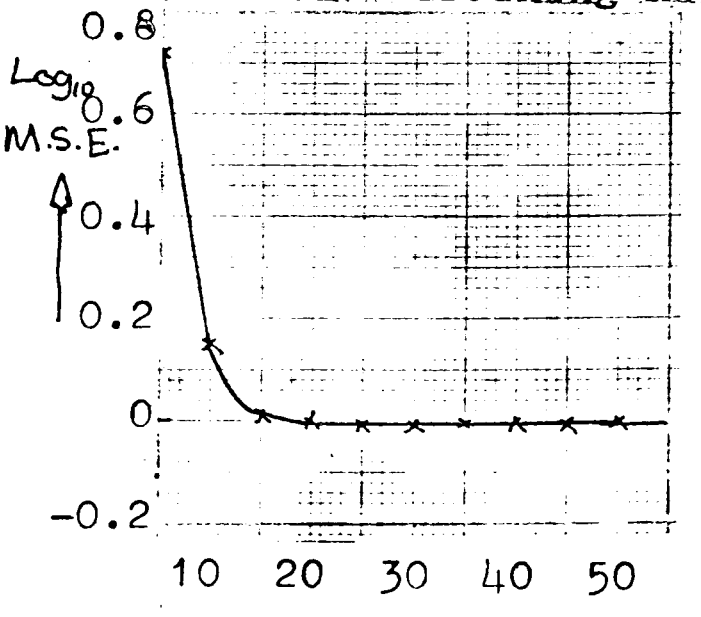
f(x) = 10*Sin(x) Horizontal axes - Training Runs.





f(x) = 1.1*Sin(x) Horizontal axes - Training Runs





f(x) = Sin(x) + 1.0 Horizontal axes - Training Runs.

Figure 2.2.3 Mean Square Error minimising for f(x)=Sin(x)+1.0.

SIN(X)	COEFFICIENTS	
60^{o} samples.	-1	1.000
30^{o} "	-1	1.73205
15^{o} "	-1	1.93185
10^{o} "	-1	1.96924
1^{o} "	-1	1.999619
0.5^{o} "	-1	1.999924
RAMP	-1	2

SINH(X)	COEFFICIENTS	
Sampled Every		
4^{o}	-1	54.616
2^{o}	-1	7.524
1^{o}	-1	3.08616
0.5^{o}	-1	2.2552
0.3^{o}	-1	2.0906
0.2^{o}	-1	2.0401
0.1^{o}	-1	2.0100

Table 2.2.1 Relationship between coefficients and simple waveforms.

DATA	FORECAST	TRAINING RUN	MEAN SQUARE ERROR	LOG. M.S.E.
0.0	-			
0.50000	-	5	$5.01*10^{0}$	0.699
0.86602	0.86602	10	$4.31*10^{-1}$	-0.365
1.00000	1.00000	15	$3.72*10^{-2}$	-1.430
0.86602	0.86602	20	$3.20*10^{-3}$	-2.494
0.50000	0.50000	25	$2.76*10^{-4}$	-3.559
0.0	0.0	30	$2.38*10^{-5}$	-4.623
-0.50000	-0.50000	35	$2.05*10^{-6}$	-5.688
-0.86602	-0.86602	40	$1.77*10^{-7}$	-6.752
-1.00000	-1.00000	45	$1.52*10^{-8}$	-7.817
-0.86602	-0.86602	50	$1.31*10^{-9}$	-8.882
-0.50000	-0.50000	55	$1.13*10^{-10}$	-9.946
0.0	0.0	60	$9.76*10^{-12}$	-11.011

$f(x) = Sin(x)$ sampled every 30^{o}

$C_2 = -1.00000$ $C_1 = 1.73205$

DATA	FORECAST	TRAINING RUN	MEAN SQUARE ERROR	LOG. M.S.E.
0.0	-			
5.00000	-	5	$5.01*10^{2}$	2.699
8.66025	8.66025	10	$4.31*10^{1}$	1.635
10.00000	10.0000	15	$3.72*10^{0}$	0.570
8.66025	8.66025	20	$3.20*10^{-1}$	-0.494
5.00000	5.00000	25	$2.76*10^{-2}$	-1.559
0.0	0.0	30	$2.38*10^{-3}$	-2.623
-5.00000	-5.00000	35	$2.05*10^{-4}$	-3.688
-8.66025	-8.66025	40	$1.77*10^{-5}$	-4.752
-10.0000	-10.0000	45	$1.52*10^{-6}$	-5.817
-8.66025	-8.66025	50	$1.31*10^{-7}$	-6.882
-5.00000	-5.00000	55	$1.13*10^{-8}$	-7.946
0.0	0.0	60	$9.76*10^{-10}$	-9.011

$f(x) = 10*Sin(x)$ sampled every 30^{o}

$C_2 = -9.99999$ $C_1 = 17.32050$

Table 2.2.2 Training coefficients for Sin(x) and 10*Sin(x).

DATA	FORECAST	TRAINING RUN	MEAN SQUARE ERROR	LOG. M.S.E.
1.00000	-	5	$5.79*10^{0}$	0.763
6.00000	-	10	$1.40*10^{0}$	0.145
9.66025	8.48704	15	$1.02*10^{0}$	0.008
11.00000	9.90000	20	$9.85*10^{-1}$	-0.007
9.66025	8.66025	25	$9.82*10^{-1}$	-0.008
6.00000	5.10000	30	$9.82*10^{-1}$	-0.008
1.00000	0.17321	35	$9.82*10^{-1}$	-0.008
-4.00000	-4.80000	40	$9.82*10^{-1}$	-0.008
-7.66025	-8.48704	45	$9.82*10^{-1}$	-0.008
-9.00000	-9.90000	50	$9.82*10^{-1}$	-0.008
-7.66025	-8.66025	55	$9.82*10^{-1}$	-0.008
-4.00000	-5.10000	60	$9.82*10^{-1}$	-0.008
1.00000	-0.17321			

$f(x) = Sin(x) + 1.0$ sampled every 30^{o}

$C_2 = -0.96000$ $C_1 = 1.69741$

DATA	FORECAST	TRAINING RUN	MEAN SQUARE ERROR	LOG. M.S.E.
0.0	-			
0.55000	-	5	$6.06*10^{0}$	0.782
0.95263	0.95263	10	$5.22*10^{-1}$	-0.282
1.10000	1.10000	15	$4.50*10^{-2}$	-1.347
0.95263	0.95263	20	$3.88*10^{-3}$	-2.411
0.55000	0.55000	25	$3.34*10^{-4}$	-3.476
0.0	0.0	30	$2.88*10^{-5}$	-4.541
-0.55000	-0.55000	35	$2.48*10^{-6}$	-5.605
-0.95263	-0.95263	40	$2.14*10^{-7}$	-6.670
-1.10000	-1.10000	45	$1.84*10^{-8}$	-7.734
-0.95263	-0.95263	50	$1.59*10^{-9}$	-8.799
-0.55000	-0.55000	55	$1.37*10^{-10}$	-9.863
0.0	0.0	60	$1.18*10^{-11}$	-10.928

$f(x) = 1.1*Sin(x)$ sampled every 30^{o}

$C_2 = -1.10000$ $C_1 = 1.90526$

Table 2.2.3 Training coefficients for Sin(x) + 1.0 and 1.1*Sin(x).

Scale	Mean Square Error	Training Runs	Coefficients
100	$0.69690*10^{-4}$	24	-0.97639 1.99
	$0.44601*10^{-4}$	25	-0.98111 1.99
	$0.28545*10^{-4}$	26	-0.98489 1.99
	$0.18269*10^{-4}$	27	-0.98791 2.00
	$0.11692*10^{-4}$	28	-0.99033 2.00
	$0.74829*10^{-5}$	29	-0.99226 2.00
	$0.47890*10^{-5}$	30	-0.99381 2.00
	$0.30650*10^{-5}$	31	-0.99505 2.00
	$0.19616*10^{-5}$	32	-0.99604 2.00
	$0.12554*10^{-5}$	33	-0.99683 2.00
10	$0.64904*10^{-5}$	19	-0.92794 1.97
	$0.41538*10^{-5}$	20	-0.94235 1.98
	$0.26585*10^{-5}$	21	-0.95388 1.98
	$0.17014*10^{-5}$	22	-0.96311 1.99
	$0.10889*10^{-5}$	23	-0.97049 1.99
	$0.69690*10^{-6}$	24	-0.97639 1.99
	$0.44601*10^{-6}$	25	-0.98111 1.99
	$0.28545*10^{-6}$	26	-0.98489 1.99
	$0.18269*10^{-6}$	27	-0.98910 2.00
	$0.11692*10^{-6}$	28	-0.99033 2.00
	$0.74829*10^{-7}$	29	-0.99226 2.00
	$0.47980*10^{-7}$	30	-0.99381 2.00
	$0.30650*10^{-7}$	31	-0.99505 2.00
	$0.19616*10^{-7}$	32	-0.99604 2.00
1	$0.87961*10^{-5}$	8	-0.16114 1.66
	$0.56295*10^{-5}$	9	-0.32891 1.73
	$0.36029*10^{-5}$	10	-0.46313 1.79
	$0.23058*10^{-5}$	11	-0.57050 1.83
	$0.14757*10^{-5}$	12	-0.65640 1.86
	$0.94447*10^{-6}$	13	-0.72512 1.89
	$0.60446*10^{-6}$	14	-0.78010 1.91
	$0.38686*10^{-6}$	15	-0.82408 1.93
	$0.24759*10^{-6}$	16	-0.85926 1.94
	$0.15846*10^{-6}$	17	-0.88741 1.95
	$0.10141*10^{-6}$	18	-0.90993 1.96
	$0.64904*10^{-7}$	19	-0.92794 1.97
	$0.41538*10^{-7}$	20	-0.94235 1.98
	$0.26585*10^{-7}$	21	-0.95388 1.98
	$0.17014*10^{-7}$	22	-0.96311 1.99
	$0.10889*10^{-7}$	23	-0.97049 1.99
	$0.69690*10^{-8}$	24	-0.97639 1.99
	$0.44601*10^{-8}$	25	-0.98111 1.99
	$0.28545*10^{-8}$	26	-0.98489 1.99
	$0.18269*10^{-8}$	27	-0.98791 2.00
	$0.11692*10^{-8}$	28	-0.99033 2.00
	$0.74829*10^{-9}$	29	-0.99226 2.00
	$0.47890*10^{-9}$	30	-0.99381 2.00
	$0.30650*10^{-9}$	31	-0.99505 2.00
	$0.19616*10^{-9}$	32	-0.99604 2.00
	$0.12554*10^{-9}$	33	-0.99680 2.00

Ramp Data Scaled by 1, 10 and 100.

Table 2.2.4 Training coefficients for a Ramp function.

Sampled every	Mean Square Error	Training Runs	Coefficients
1^o	$0.21161*10^{-7}$	24	-0.97643 1.99
	$0.13541*10^{-7}$	25	-0.98114 1.99
	$0.86653*10^{-8}$	26	-0.98492 1.99
	$0.55451*10^{-8}$	27	-0.98793 1.99
	$0.35484*10^{-8}$	28	-0.99035 1.99
	$0.22707*10^{-8}$	29	-0.99228 2.00
	$0.14531*10^{-8}$	30	-0.99382 2.00
	$0.92986*10^{-9}$	31	-0.99506 2.00
	$0.59504*10^{-9}$	32	-0.99605 2.00
	$0.38078*10^{-9}$	33	-0.99684 2.00
$\frac{1}{2}^o$	$0.53029*10^{-8}$	24	-0.97640 1.99
	$0.33938*10^{-8}$	25	-0.98112 1.99
	$0.21719*10^{-8}$	26	-0.98490 1.99
	$0.13900*10^{-8}$	27	-0.98792 2.00
	$0.88958*10^{-9}$	28	-0.99033 2.00
	$0.56931*10^{-9}$	29	-0.99227 2.00
	$0.36435*10^{-9}$	30	-0.99381 2.00
	$0.23318*10^{-9}$	31	-0.99505 2.00
	$0.14923*10^{-9}$	32	-0.99604 2.00
	$0.95503*10^{-10}$	33	-0.99683 2.00
$\frac{1}{4}^o$	$0.13265*10^{-8}$	24	-0.97639 1.99
	$0.84897*10^{-9}$	25	-0.98111 1.99
	$0.54334*10^{-9}$	26	-0.98489 1.99
	$0.34773*10^{-9}$	27	-0.98791 2.00
	$0.22255*10^{-9}$	28	-0.99033 2.00
	$0.14243*10^{-9}$	29	-0.99226 2.00
	$0.91154*10^{-10}$	30	-0.99381 2.00
	$0.58338*10^{-10}$	31	-0.99505 2.00
	$0.37336*10^{-10}$	32	-0.99604 2.00
	$0.23895*10^{-10}$	33	-0.99683 2.00

Sinewave Sampled at 1^o, $\frac{1}{2}^o$ and $\frac{1}{4}^o$.

Table 2.2.5 Relationship between oversampled Sin(x) and a Ramp function.

2.3 Investigation of exponential constraints

Program A2 had limitations in that it could only deal with two linear coefficients. Modifications were carried out to enable an unlimited number of linear coefficients to be used and also to increase its speed. The modified version is shown in Appendix A, program A3. The next stage in the investigation was an attempt to find an association between the number of linear coefficients and some component of the waveform; this turned out to be of an exponential nature. Also an investigation was undertaken to explore the possibility of generating various waveforms by changing the coefficient values and their corresponding starting values.

From the theory, it can be shown that the number of linear prediction coefficients required to adequately represent a waveform is equal to the number of real and complex exponential components related to that waveform. This gives some insight into the problem of choosing the optimum number of coefficients to represent a given waveform.

Program A3 can deal with an unlimited number of linear coefficients. Table 2.3.1 shows that, for a single sinewave sampled at every 30°, the coefficients are -1.0 and 1.73205. As in the previous section, these values are independent of the amplitude of the sinewave. Similarly, table 2.3.1 shows that, for two sinewaves added together and with $(\sin x + \sin 3x)$ sampled every 30°, the coefficients are -1.0, 1.73205, -2.0 and 1.73205. A problem that can arise is that of over sampling the waveform, say, every degree. Although the waveform is

composed of two sinewaves of different frequencies, if only the first 10 points are used, the waveform approaches that of two ramps added together i.e. a single ramp which only needs two linear coefficients as shown in table 2.3.2. If more coefficients are used than are actually required another problem develops, some of the extra coefficients have arbitrary values independent of the waveform and the remaining coefficients are dependent on them, consequently the rate of convergence onto a set optimum values decreases. Also because some of the coefficients are arbitrary there is an infinite set of solutions this is shown in table 2.3.1. For the same number of training runs 2,000 and similar times approximately 9 seconds, the error for four independant coefficients, drops to within the machines limit, less than 10^{-26} but for inter-dependent coefficients, that is, using 4 coefficients where only 2 are necessary the error is only 10^{-14}. Obviously this error is small enough for most practical purposes but it demonstrates the slower convergence rate. In fact the error for the independent coefficients reaches 10^{-26} in less than 50 training runs.

The experiments in section 2.1 using a ramp function as input data can now be reconsidered in the knowledge that two linear coefficients give an exact solution for a ramp. Thus, using more coefficients results in a polynomial with interdependent coefficients. We must thus re-examine the conclusions drawn in section 2.1. Results shown in tables 2.1.1 and 2.1.2 can be ignored because they were obtained using a different criteria to stop the training of the coefficients. However, for a mean square error of less than 0.1, 0.01 and 0.001, coefficients of correlations between the number of training

runs and the number of terms in the polynomial are -0.0584, 0.3155 and 0.4498 respectively, the implication being that for a small error there is no relationship, and as the error becomes smaller some relationship appears to form. Bearing in mind that the polynomial goes from an optimum setting of two coefficients to a maximum of 55 in all (5 linear, 15 quadratic, and 35 cubic terms), the inter-dependence between terms can have a large affect on any corresponding correlation tests. Thus, if a simple sinewave is added to another sinewave of three times the frequency, and also if 4 linear coefficients, which exactly fit the waveform are used, the measurements given in table 2.3.1 can be taken, for a waveform sampled every 25.7^{o}. After 50 training runs the error was only 0.45×10^{-2} for the 4 coefficients -0.812, 1.964, -2.541 and 2.082 whose true values were -1.0, 2.247, -2.802 and 2.247 respectively. Although the rule of thumb that states that, for m coefficients, $\frac{1}{2}m^2$ training runs would be sufficient to bring the coefficients to within a small fraction of their optimum value, the matter is obviously subjective and depends upon how small a 'small fraction' is. After 50 training runs, instead of $\frac{1}{2} \times 4 \times 4$, i.e. 8 training runs, the coefficients obtained above differ from their true values by as much as 18.8% and by as little as 7%.

The rate of convergence, even for such deterministic systems depends upon the sampling rate. If we consider (sin x + sin 3 x) again, but sampled every 36^{o}, the coefficients quickly converge to -1.0, 1.0, -1.0 and 1.0 in less than 50 training runs. The experiment becomes easier if carried out with a single sinewave as shown in table 2.3.1, and using the results of (sin x) sampled every 30^{o}. If we compare the graphs in figure 2.3.1 of the coefficients converging to their optimum values, it can be seen that they converge at a similar rate, and that

it takes approximately 12 training runs before the coefficients come to within 10% of the optimum value. Figure 2.2.3 shows that the relationship between the number of training runs and the mean square error is a logarithmic one, except when a less-than-optimum number of coefficients is used, when it reaches a minimum, as in the case of (sin x + 1.) which required more than 20 training runs to reach a minimum. Three linear coefficients are required for this waveform. Table 2.3.3 a and b shows similar results using a ramp function. A noticeable difference being that for the data 1 to 10, after 1000 training runs, the coefficients are -0.993 and 1.994 (less than 1% from their optimum values of -1.000 and 2.000) and that the error is reduced to 0.74×10^{-5}. The convergence rate thus seems greatly affected, not only by the size of the history (input data), but also by its content. Convergence can be increased by reducing the number of data points to four. Table 2.3.4 shows that for the 4 points 1 to 4, after the same amount of training, the error is less than 10^{-13}; with the data set changed to 0 to 3, the error decreases to less than 10^{-26}; and with the data set -1 to 2, the error decreases to zero. If, however, the data is shifted one more place to the left giving to set -2 to 1, the error increases to 0.154×10^{-26} after 1000 training runs. In fact, for the data set -1,0, 1 and 2, the error is zero after the first training run. The reason for the variation of rates of convergence is explained in chapter 3 section 3.3.

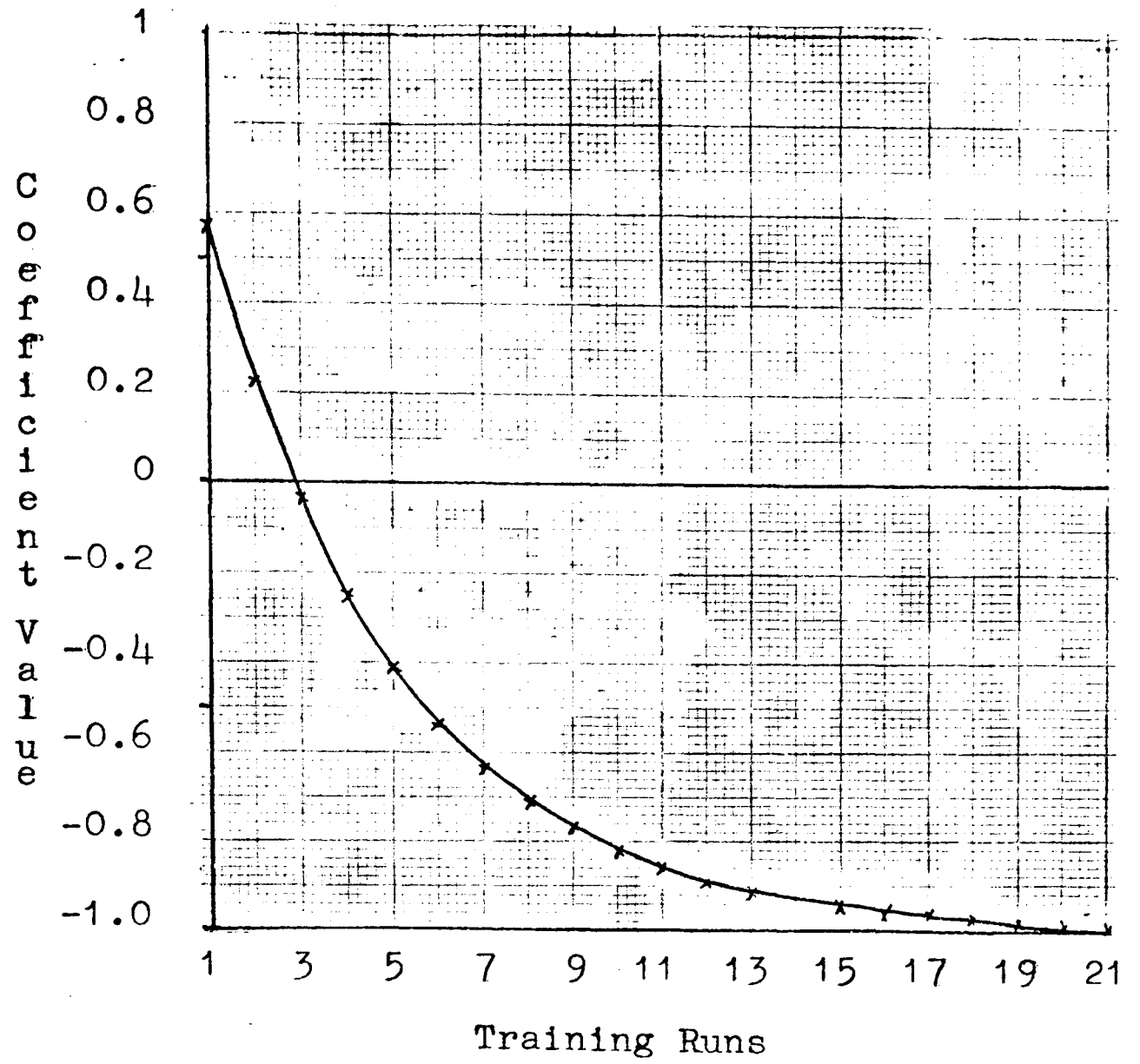
Another program was written to investigate the different types of waveforms that could be generated by linear prediction; these are shown in figure 2.3.2 a and b and are of the form $e^{\pm ax} \sin bx$. Again, only two linear coefficients are needed because they only contain two

exponential components. The rate of convergence depends upon whether the absolute value of C_2 is greater or smaller than -1. It is evident at this stage that both coefficients are no longer independent as assumed previously, and where C_1 was previously affected by a change in the frequency of the sinewave or by a different sampling rate, given that C_2 is greater or smaller than the critical value of -1, C_1 also has an effect on the rate of convergence (or divergence) : the mathematical explanation of this is given in chapter 3 section 3.3.

Additional work was subsequently aimed at the generation of waveforms of the types shown in figures 2.3.3 a and b. Using coefficient values of C_1 and C_2 both equal to 0.4, and repeated using 0.5, the two waveforms generated appear to disprove the theory of an association between the number of linear coefficients and the exponential components of the waveforms : in the case of C_1 and C_2 equal to 0.4, the waveform appears to be that of an exponentially-damped sinewave plus a d.c. level, the damped sinewave requiring two linear coefficients and the d.c. level one, and totalling three coefficients in all. This problem arises because the two starting values are chosen at random, and are in these cases 0 and 1. This is the main reason for the anomaly, and also as both coefficients are positive, for two positive starting values, the result must also be positive. A mathematical explanation is given in chapter 3 section 3.3.

Another program was written to generate not only linear, but also quadratic and cubic terms; similar problems occur in the arbitrary choice of the acceptable starting values and their corresponding

coefficient values, and also in that the number of variables rapidly increases and the majority of waveforms generated diverge at very high rates. This is illustrated in figure 2.3.3 c for the case of a quadratic coefficient greater than 1; it is evident just how quickly it diverges. Similarly, for two quadratic coefficients both slightly greater than 1, their speed of convergence at an even greater rate is shown in figure 2.3.3 c. This haphazard random selection of starting values and coefficient values is unsatisfactory because there is no assurance that an optimum selection has been made.



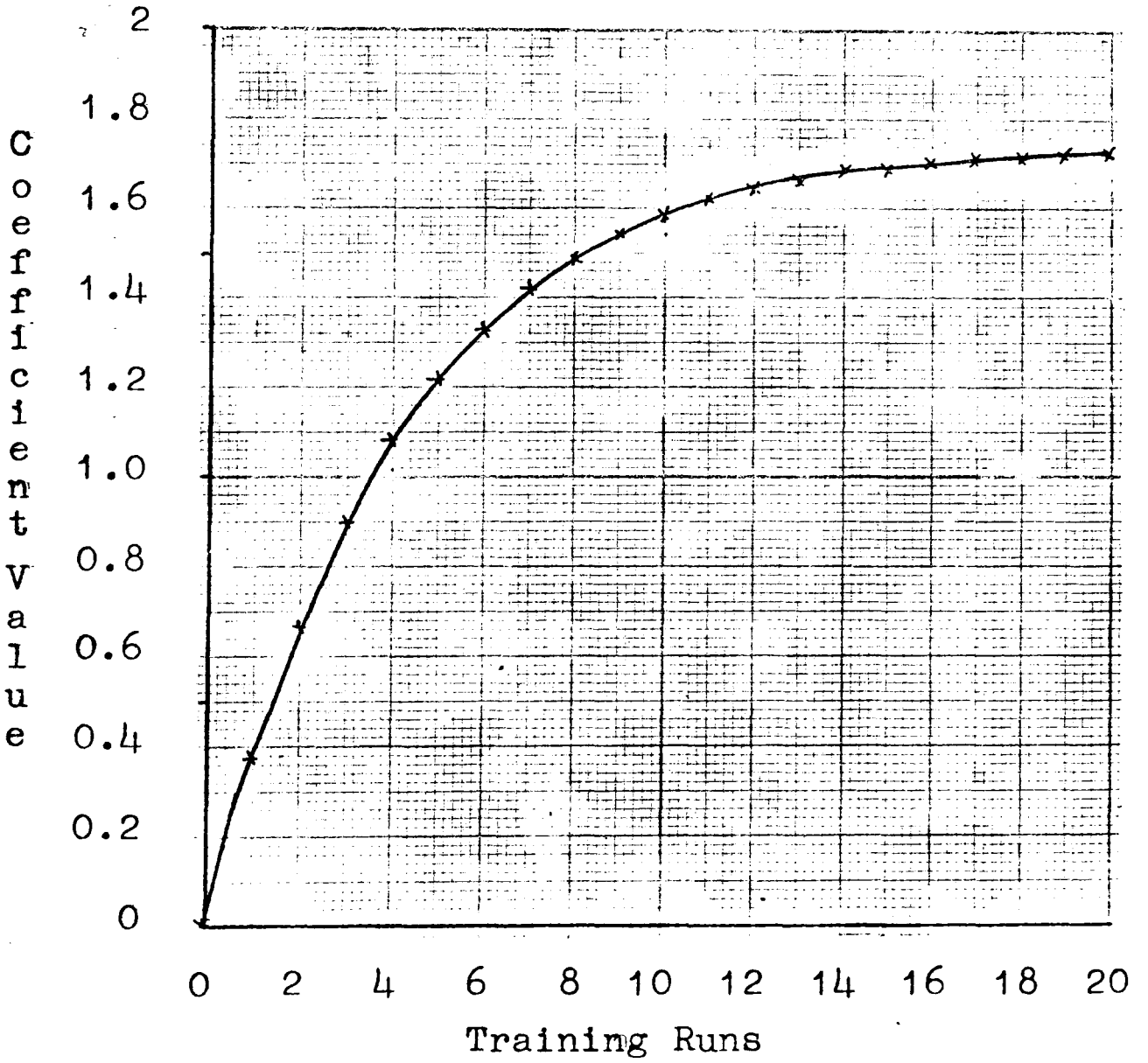
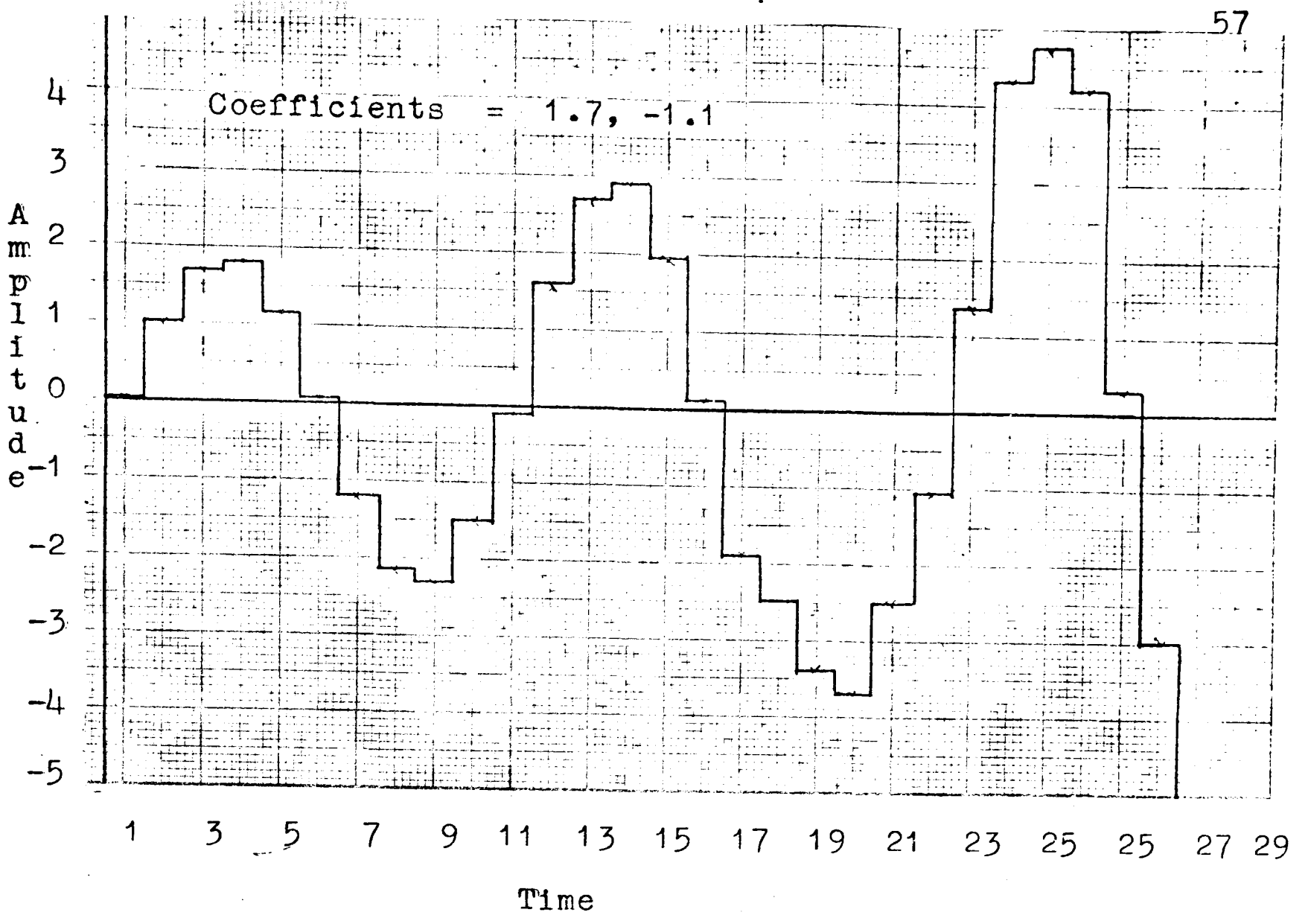
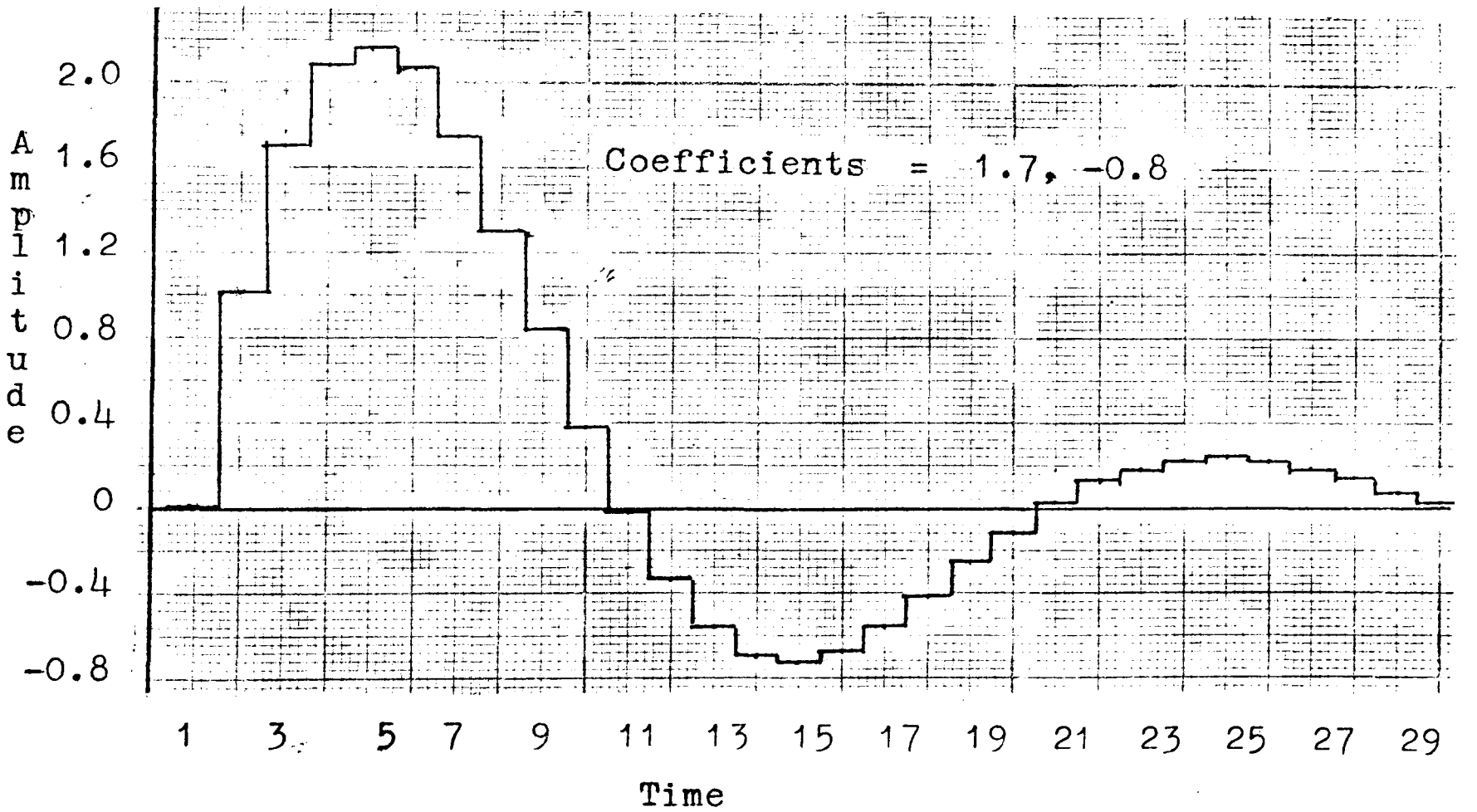


Figure 2.3.1 Convergence rate of coefficients.



Starting values 0.0, 1.0.

Figure 2.3.2a Generating exponentially increasing sinewave with two coefficients.



Starting values 0.0, 1.0.

Figure 2.3.2b Generating exponentially damped sinewave with two coeffcients.

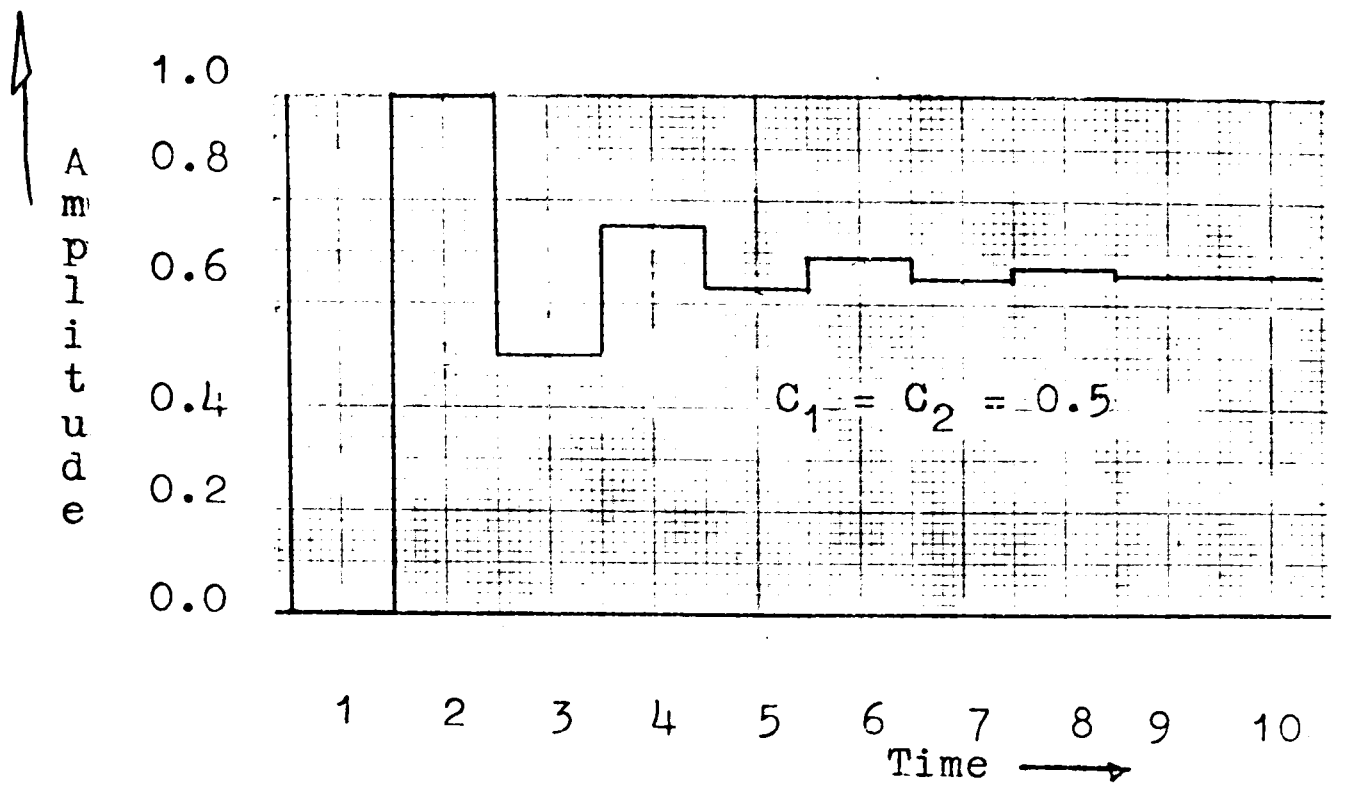


Figure 2.3.3a Generation of complex waveform with two coefficients.

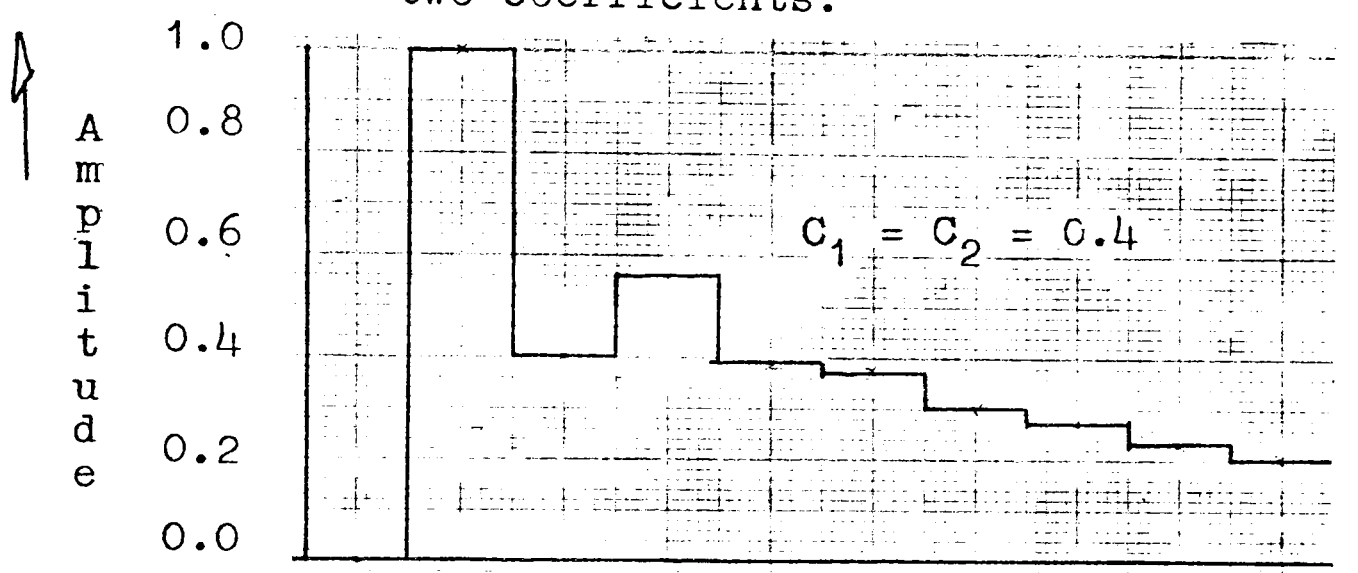


Figure 2.3.3b Generation of complex waveform with two coefficients.

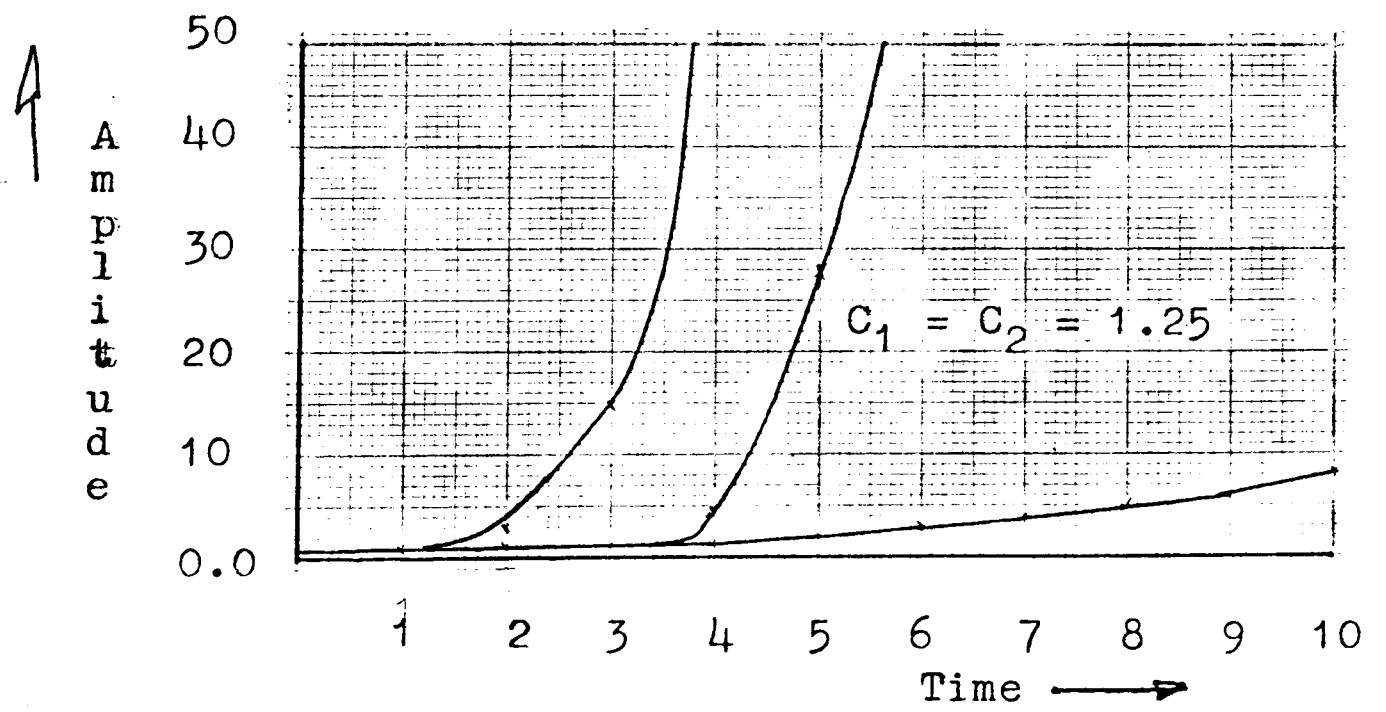


Figure 2.3.3c Divergence rates of quadratic terms in the polynomial.

Sampled Every	Mean Square Error.	Number of Training Runs.	Coefficients.
30^o	10^{-20}	1,000	-1.0 1.732
0.5^o	$0.1346*10^{-4}$	50	-1.330 1.333 0.673 0.380
	$0.2986*10^{-14}$	2,000	-1.499 1.395 0.704 0.397
25.7^o	$0.4887*10^{-2}$	50	-0.812 1.964 2.541 2.082
	$0.3824*10^{-23}$	2,000	-1.0 2.247 -2.802 2.247
30^o	$0.1021*10^{-5}$	50	-0.998 1.730 -1.998 1.731
	$0.2304*10^{-24}$	1,000	-1.00 1.732 -2.00 1.732
36^o	$0.1824*10^{-26}$	50	-1.00 1.00 -1.00 1.00
	$0.1824*10^{-26}$	2,000	-1.00 1.00 -1.00 1.00

Upper Section:- $\sin(x)$ Sampled at 30^o using two coefficients. Sampled at 0.5^o using four coefficients. Thus over specified (only two coefficients required) solution obtained is one of an infinite set.

Lower Section:- $\sin(x) + \sin(3x)$ Sampled at 25.7^o, 30^o and 36^o respectively.

Table 2.3.1 Effect of a non-optimum number of coefficients.

Sampled at Intervals	F(x) = Sin(x) + Sin(3x). Coefficients.	Number of Training Runs.	Mean Square Error.
0.5^{o}	0.309 0.916 -1.000 1.998	50 1,000	$0.176*10^{-2}$ $0.786*10^{-12}$
0.5^{o}	-1.334 1.325 0.673 0.383 -1.492 1.382 0.702 0.399	50 1,000	$0.469*10^{-4}$ $0.303*10^{-11}$

Upper Section:- Oversampled therefore approximates to a ramp (thus coefficients tend to -1.0 2.0).

Lower Section:- Using correct number of coefficients but data oversampled (that is, approximate to a ramp).

Table 2.3.2 Effect of oversampling in the time domain waveforms.

Number of Data Points.	Mean Square Error.	No. of Training Runs.	Coefficients. (Data Equals a Ramp).	
20 (5.78secs.)	$0.263*10^{-3}$	50	0.930	0.224
	$0.285*10^{-5}$	1,000	-0.799	1.815
13 (3.66secs.)	0.605	50	0.887	0.337
	$0.545*10^{-3}$	1,000	-0.943	1.950
10 (2.81secs.)	0.466	50	0.769	0.505
	$0.739*10^{-5}$	1,000	-0.993	1.994
8 (2.36secs.)	0.339	50	0.636	0.682
	$0.449*10^{-7}$	1,000	-0.999	2.00
6 (1.74 secs.)	0.197	50	-0.450	0.925
	$0.111*10^{-10}$	1,000	-1.00	2.00
4 (1.2 secs.)	$0.192*10^{-12}$	50	-1.000	2.000
	$0.477*10^{-28}$	1,000	-1.000	2.000

Figure in Seconds Refers to Computation Time.

Table 2.3.3a Convergence rates of small data groups.

Data.	Forecast.	Training Runs.	Mean Square Error.	Log. M.S.E.
0	—	5	$0.109*10^{-1}$	-1.964
0.1	—			
0.2	0.06152	10	$0.102*10^{-1}$	-1.991
0.3	0.18130			
0.4	0.30109	15	$0.960*10^{-2}$	-2.018
0.5	0.42087			
0.6	0.54065	20	$0.902*10^{-2}$	-2.045
0.7	0.66043			
0.8	0.78022	25	$0.847*10^{-2}$	-2.072
0.9	0.90000			
1.0	1.01978	30	$0.796*10^{-2}$	-2.099
1.1	1.13957			
1.2	1.25935	35	$0.748*10^{-2}$	2.126

$C_1 = 0.62382, \quad C_2 = 0.58261.$

After 35 training runs mean square error is $0.748*10^{-2}$ and the coefficients have not reached their optimum.

Table 2.3.3b Convergence rate for Ramp data.

Sets of Four Data Values.	Mean Square Error	No. of Training Runs.	Coefficients.		Computation. Time. (Seconds)
2 3 4 5	0.114	50	1.381	0.286	
	$0.326*10^{-3}$	1,000	-0.873	1.908	1.154
1 2 3 4	$0.875*10^{-1}$	50	0.497	1.079	
	$0.141*10^{-13}$	1,000	-1.00	2.00	1.141
0 1 2 3	$0.636*10^{-9}$	50	-1.00	2.00	
	$0.858*10^{-26}$	1,000	-1.00	2.00	1.135
-1 0 1 2	0.0	50	-1.00	2.00	
	0.0	1,000	-1.00	2.00	1.107
-2 -1 0 1	$0.127*10^{-9}$	50	-1.00	2.00	
	$0.154*10^{-26}$	1,000	-1.00	2.00	1.133
-3 -2 -1 0	$0.337*10^{-1}$	50	-0.424	1.079	
	$0.542*10^{-14}$	1,000	-1.00	2.00	1.148

Increased Convergence by using different Data Seta.

Table 2.3.4 Effect of different data sets on convergence.

CHAPTER THREE

3 THEORY

3.1 Theory of Learning Program

Learning, used in this sense, is intended to mean the self-adjustment of a set of variable parameters by some error measuring criteria, such that the product resulting minimises the difference between it and a desired product.

In his optimisation of a universal non-linear filter, simulator and predictor, Gabor uses this definition in a system which is described by a polynomial. The polynomial contains a set of groups, the number of which depends on the complexity of the system, and it is constructed from the input data to the system.

A general arrangement of a cybernetic system is given in figure 3.1.1 and which may be considered as having input data in the form of a sampled waveform, a data string, or a set of numbers. An example of typical input data is given in figure 3.1.2. The first group of terms in the polynomial as shown in figure 3.1.3, contains n linear terms, each having an associated coefficient. The second group, also shown in figure 3.1.3, contain quadratic terms which are combinations of any two linear terms (as $f_1 * f_2$ is the same as $f_2 * f_1$, the lower triangle of the matrix in figure 3.1.3 is ignored). Thus, for n linear terms, a maximum of $(n + 1) * n/2$ quadratic terms are available, each having an associated coefficient. Cubic terms, as given in figure 3.1.4, are formed by any combination of three linear

terms, and give a maximum of $(n + 2)(n + 1)*n/6$ terms, each with a coefficient. The series continues similarly for higher order terms. For all practical purposes, only linear, quadratic and cubic groups are used because, for a reasonable value of n, the polynomial becomes excessively large and requires considerable computer time.

The flow chart for programs A1, A2 and A3 (Appendix A) is illustrated in figure 3.1.5. Given one of these programs and a value for n chosen for a set of suitable data, the program can cyclically adjust the coefficients in such a manner as to minimise the difference between the polynomial f(0) and some desired valve g(0), in correspondence to a given criterion. The criterion chosen is the minimisation of the mean square error:

$$f(0) = \sum_{j}^{n} C_j f_i + \sum_{j}^{n} \sum_{i}^{n} C_{ji} f_j f_i + \sum_{k}^{n} \sum_{j}^{n} \sum_{i}^{n} C_{kji} f_k f_j f_i + \cdots \qquad 3.1.1$$

If the history is m data points long, m being greater than n (the number of linear terms) then (m - n) equations of the form shown above in equation 3.1.1 can be obtained from the history by taking a window of length n and sliding it along the m data points. The n data points viewed in this window are used to construct these equations. Thus, there are (m - n) values of f(0), and a corresponding (m - n) values of g(0). The error, derived as the difference between f(0) and g(0) is achieved by squaring, summing and averaging the differences. The goodness of fit of the instantaneous values of the coefficients is indicated by the value of the expression given in equation 3.1.2 below

$$\text{Error} = \frac{1}{m - n} \sum_{i}^{m-n} \left[g(0) \sim f(0) \right]^2 \qquad 3.1.2$$

The KOL program (program A1) is designed to scan a window of width variable between 1 and 5 across the data group, the data falling within the window being defined as a sampling group or sub-data group, and to scan through the degress of interaction i.e. the order of groups in the polynomial (for an interaction of one, only linear terms are used; for an interaction of two, both linear and quadratic groups are used, and so on). From this, the size of the polynomial can be calculated for a window size of 5 and gives 5 linear, 15 quadratic and 35 cubic terms. The highest degree of interaction used in the KOL program is three, i.e. only to cubic terms; if higher order terms are considered, the size of the polynomial rapidly increases as the window is increased.

The first application of the KOL program is the evaluation of coefficient C_i, where i ranges from 1 to the total number of terms in the polynomial. The program takes the first sub-data group from the data block and constructs the polynomial with it. Each coefficient in turn is given three arbitrary values, +1, 0 and - 1, for simplicity, and the desired value of the polynomial in each case should be equal to the next piece of data in the data block after the sub-data group. The difference or error in each case is measured. Thus there are three errors for each coefficient in each sub-data group

1) difference for C_1 when set to a value of + 1 = EB
2) " " " " 0 = EC
3) " " " " - 1 = ED

$$|EB| + |EC| + |ED| = EA \qquad 3.1.3$$

The sub-data group is then moved along the data block by one sample, the sub-data group still being maintained at the same size;

this process is repeated. A worked example is illustrated in figure 3.1.6. Once this is completed, the mean square errors are calculated by squaring the errors EA, EB, EC and ED dividing by some rationalising factor, and then by adding them.

$$EEA = \sum_{j_{m-n}}^{m-n} EA_j^2 /(m - n) \qquad 3.1.4$$

$$EEB = \sum_{j_{m-n}}^{m-n} EB_j^2 /(m - n) \qquad 3.1.5$$

$$EEC = \sum_{j_{m-n}}^{m-n} EC_j^2 /(m - n) \qquad 3.1.6$$

$$EED = \sum_{j}^{m-n} ED_j^2 /(m - n) \qquad 3.1.7$$

The maximum number of sub-data groups obtainable from the main data block is $(\bar{m} - n)$.

If the mean square error is then plotted against the separate coefficients, a multi-dimensional elliptical paraboloid (figure 3.1.7a) is produced from which it is evident, that any descending path on its surface will eventually reach a minimum value. The example shown in figure 3.1.7b is given for only one coefficient being adjusted and for which only two dimensions are shown. By plotting the mean square error as the ordinate and the arbitrary parameter values as the abscissae, for the points, (EEB, + 1), (EEC, 0) and (EED, - 1), a parabola can be constructed, and by taking the general equation of the parabola, differentiating and equating the result to zero, the substitution of the three known points, gives a minimum value for the solution. Thus the abscissae of this minimum point is the optimum value of the instantaneous value of the coefficient, and is found to be $(EE\overset{B}{C} - EED)/(2*((EE\overset{B}{C} + EED) - 2*EE\overset{C}{B})$. When all the coefficients have been adjusted once, the whole process is repeated for the second and subsequent training runs.

From the results, it is evident that either the mean square error reaches a minimum value reasonably quickly or, given enough time, decreases step by step until the minimum reached is that of the machine and is in the order of 10^{-26}. As a consequence, it is necessary to determine when the coefficient values have almost reached their optimum values.

The first method for the data group 1 to 10, the value to be predicted being 11, is that the coefficient adjustment must be stopped when the predicted or forecast value is within 10% and then 1% of 11, the predicted value. This approach, however, is not a reliably accurate test, because the mean square error is the real indicator of how close the forecasts are within the input data group. As a result a good forecast can be made for the predicted value chosen, but the same coefficients used for a different data group in the same history, e.g. 2 to 11, could result in a bad forecast.

As an alternative approach, if the mean square error can be minimised or made sufficiently small, the probability of obtaining reliable forecasts for different data groups within the same history is significantly increased. Thus for the same deterministic function, the criteria for stopping is that the mean square error has minimised, or is initially less than 0.1, and for the subsequent attempt is less than 0.01, and for the final run, less than 0.001. The method used for confirming that a minimum has been reached is to count the number of times the mean square error does not decrease after each training run. If the error does not decrease for 10 consecutive runs, it is considered as having minimised and the training is stopped.

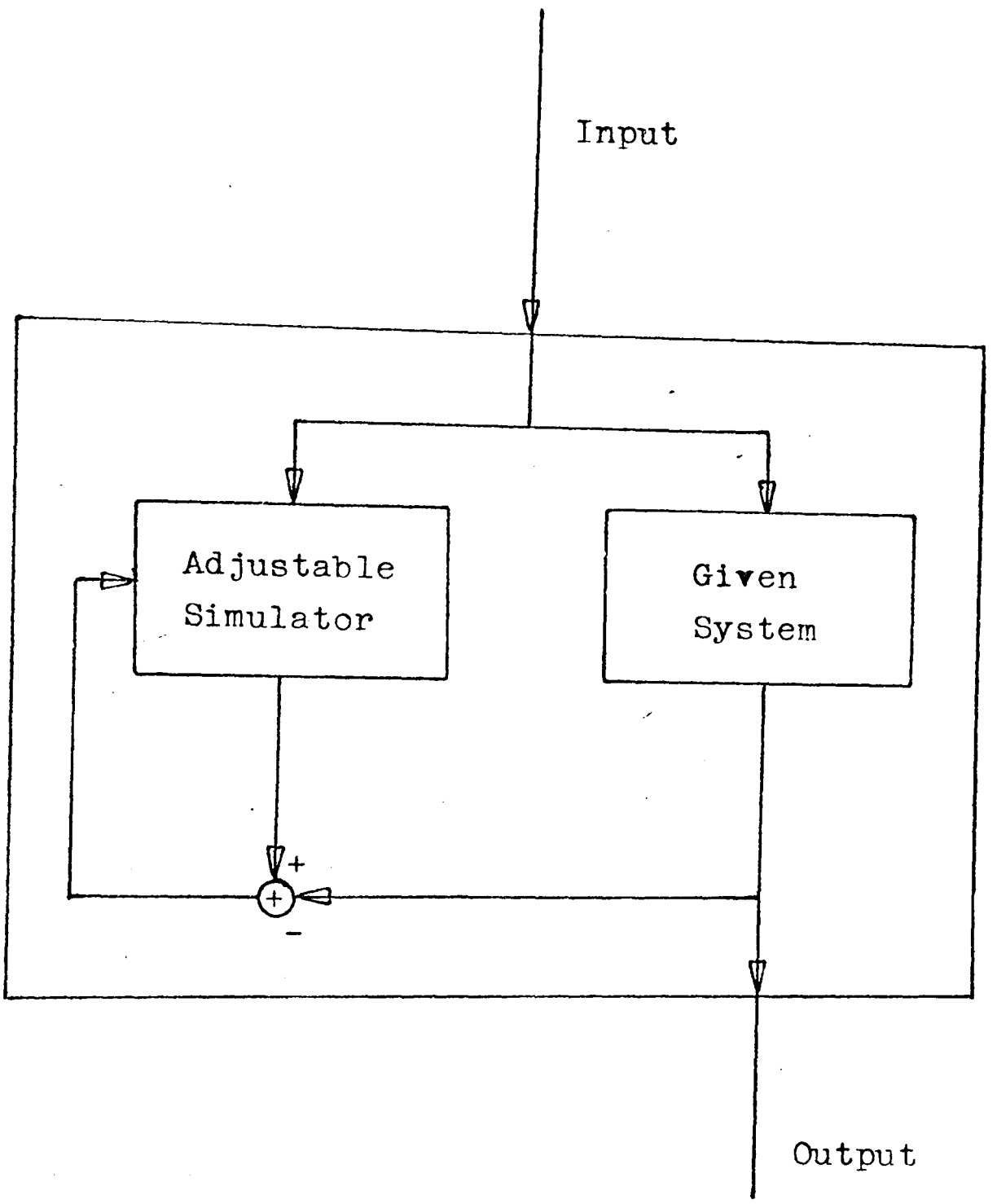
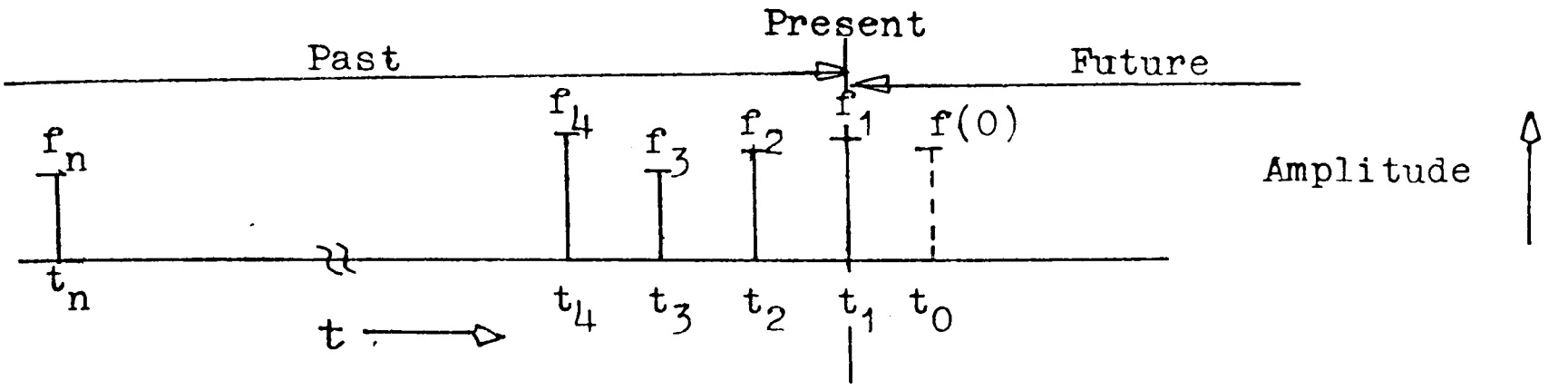


Figure 3.1.1 General Arrangement of a Cybernetic System.



Discrete Time Series.

1011, 1100, 1111, 1000, 0110, 1011, ----------

Binary String.

1.76, 2.43, 9.65, 4.79, 3.78, 6.98, ----------

Set of Numbers.

Figure 3.1.2 Example of Typical Data sets

Linear Terms -:

$$\left[C_1 f_1,\ C_2 f_2,\ C_3 f_3,\ C_4 f_4,\ \text{----------} C_n f_n \right]$$

Number of Terms -: n

Quadratic Terms -:

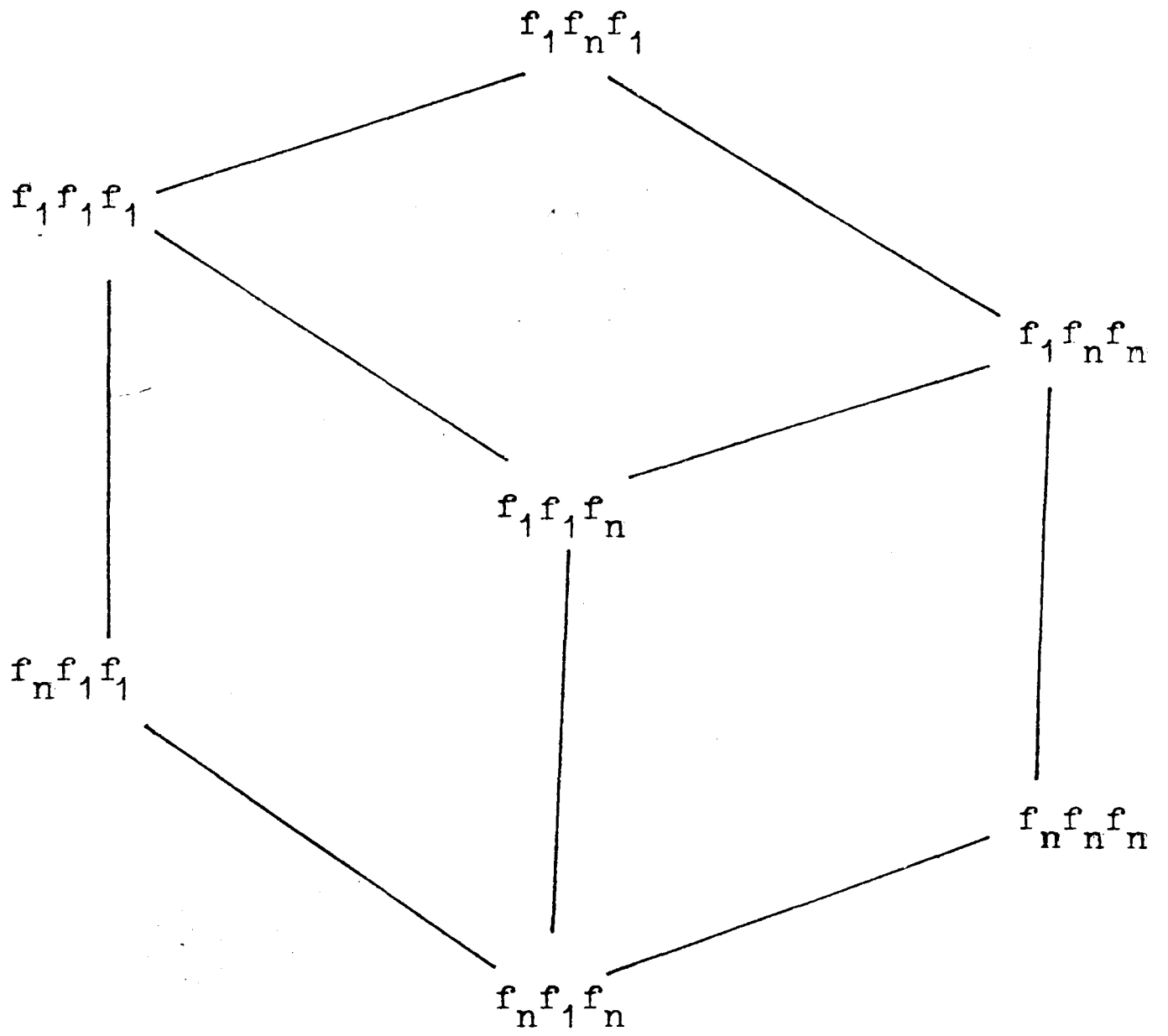
$$\begin{bmatrix} C_{11} f_1 f_1,\ C_{12} f_1 f_2,\ \text{----------} C_{1n} f_1 f_n \\ C_{21} f_2 f_1,\ C_{22} f_2 f_2, \quad - \\ - \\ \\ - \qquad\qquad - \qquad\qquad - \\ \\ \\ - \qquad\qquad\qquad - \qquad - \\ \\ C_{n1} f_n f_1,\ C_{n2} f_n f_2,\ \text{----------} C_{nn} f_n f_n \end{bmatrix}$$

$f_n f_p = f_p f_n$ etc.

Because of the terms redundancy there are infact $n(n+1)/2$ not n^2.

Figure 3.1.3 Linear and Quadratic Terms.

Cubic Terms -:



$$f_nf_mf_p = f_mf_nf_p = f_pf_mf_n \quad \text{etc.}$$

The redundancy reduces the total number of terms from n^3 to $n(n+1)(n+2)/6$.

Figure 3.1.4 Cubic Terms showing redundancy.

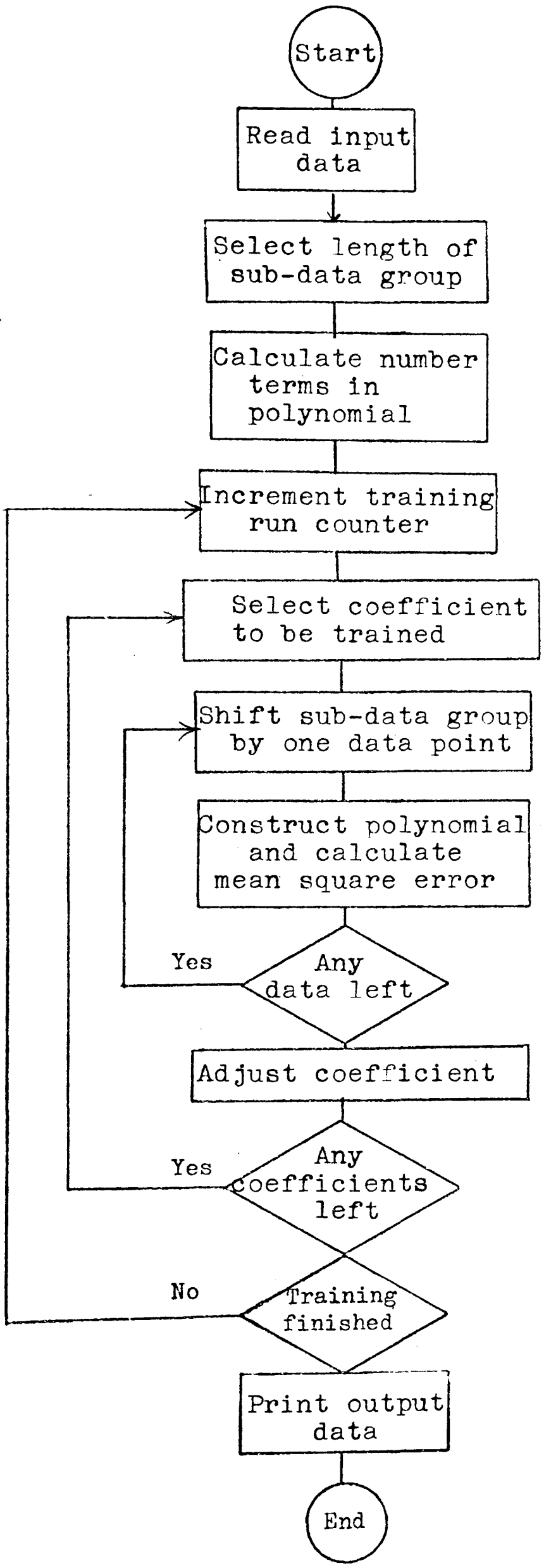


Figure 3.1.5 Functional flow chart of programs

$$S'(0) = \sum_{n}^{N} C_n f_n + \sum_{n1}^{N} \sum_{n2}^{N} C_{n1n2} f_{n1}{}^{*} f_{n2} + \text{--------}$$

Degree of interaction--- 2 (say)

Size of sub-data group-- 2 (minimum)

Data	1st. sub-data group	2nd. sub-data group
1,2,3,4,5,6	1,2	2,3

Linear terms are formed by linear combinations of the 1,2.

Quadratic terms are formed by taking all the different combinations of any two pieces of data 1,2.

Etc.

Linear terms 1,2 Quadratic terms $1^2, 1.2, 2^2$.

Cubic terms $1^3, 1^2.2, 1.2^2, 2^3$.

Thus-

$$f'(0) = C_1.1 + C_2.2 + C_3.1^2 + C_4.1.2 + C_5.2^2 + C_6.1^3 + C_7.1^2.2 + C_8.1.2^2 + C_9.1^3$$

Training run 1. $C_u = 0$ $U = 2,9$

$C_1 = -1$ $f'(0) = -1$ target value 3 EB = 4 (3--1)

$C_1 = 0$ $f'(0) = 0$ " " 3 EC = 3 (3-0)

$C_1 = +1$ $f'(0) = +1$ " " 3 ED = 2 (3-1)

EA = 9

For target value 4 (next sub-data group 2,3)

EB = 5 EC = 4 ED = 3 EA = 12

For target value 5 (next and last sub-data group 3,4)

EB = 6 EC = 5 ED = 4 EA = 15

Taking a rationalizing factor of 4

$EEB = 4^2/4 + 5^2/4 + 6^2/4 = 19.75$

$EEC = 3^2/4 + 4^2/4 + 5^2/4 = 12.50$

$EED = 2^2/4 + 3^2/4 + 4^2/4 = 7.75$

$EEA = 9^2/4 + 12^2/4 + 15^2/4 = 112.5$

First optimum adjustment to C_1

$$C_1 = (EEC - EED)/(2*((EEC+EED) - 2*EEB))$$

$$= (12.5 - 7.75)/(2*(12.5 + 7.75) - 2*19.75)$$

$$+ 4.75/(40.5 - 39.5) = 4.75$$ 2.4

Figure 3.1.6 Example of the process used in the programs.

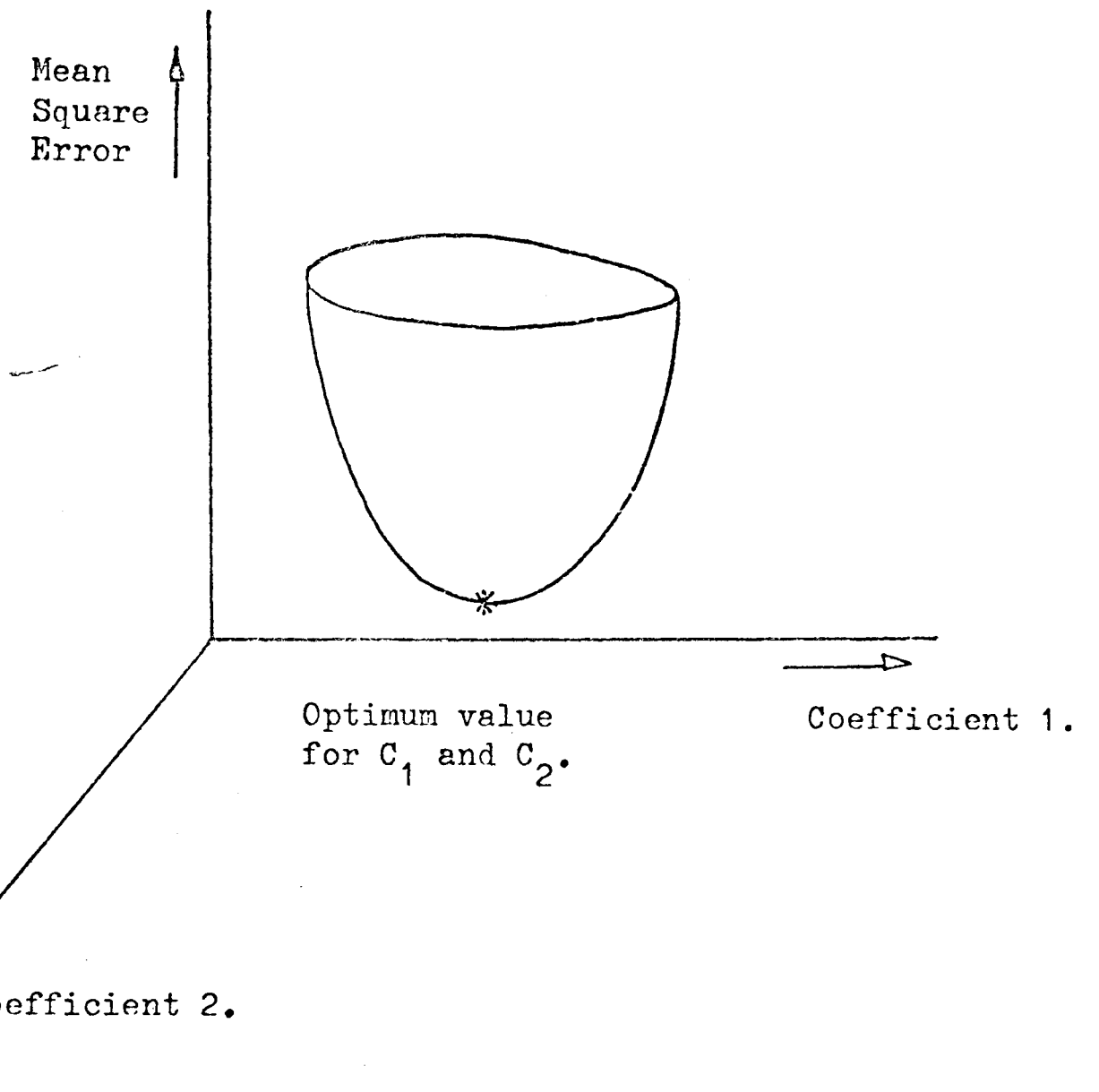


Figure 3.1.7a Relationship between two coefficients and mean square error

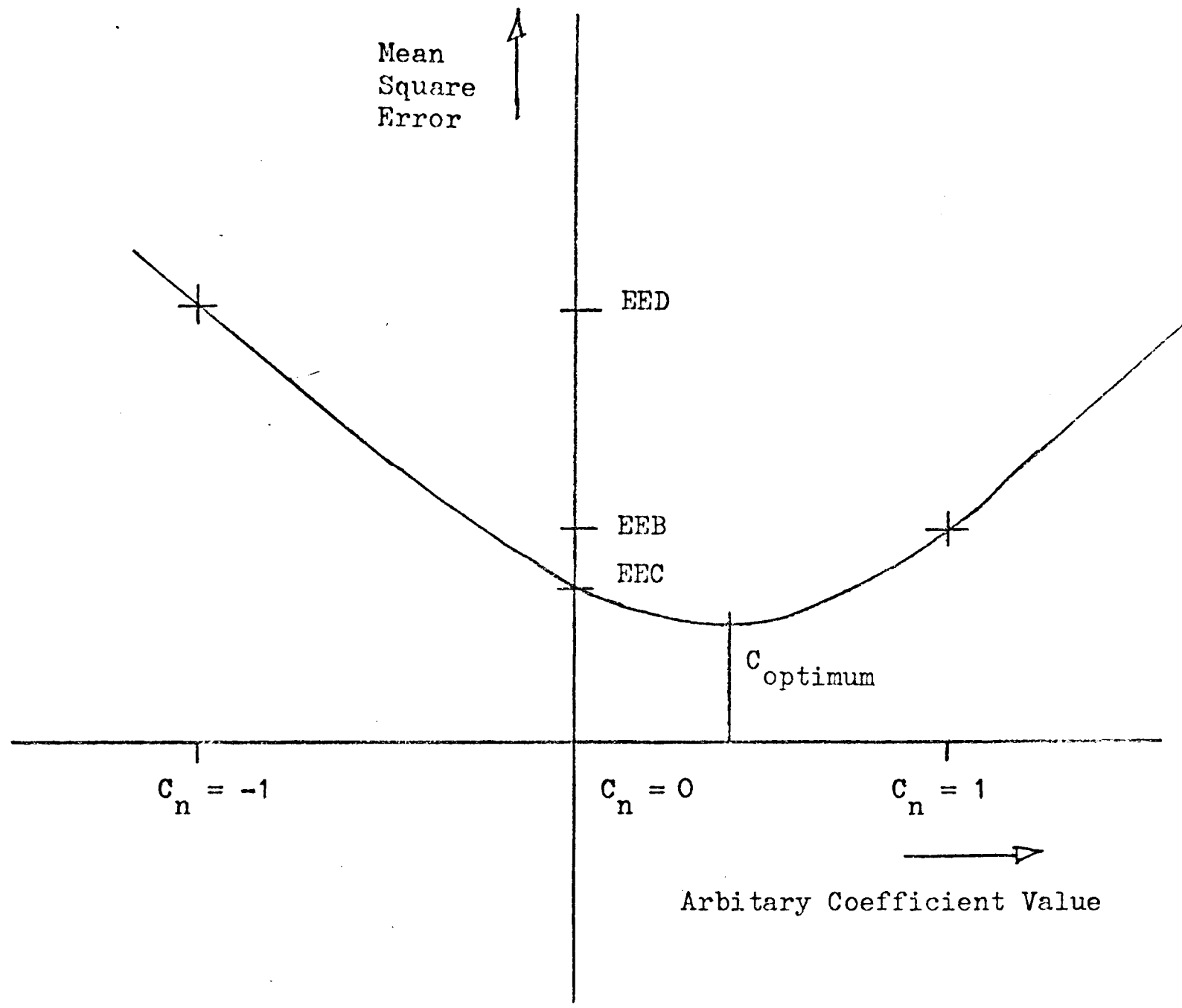


Figure 3.1.7b Relationship between optimum coefficient value and mean square error

3.2 Limitations of Learning Method

The main limitation of the learning method used in the previous section is firstly, that no method is known for choosing the size of the sub-data group (window length [20]) or the choice of the degree of interaction; i.e. whether it requires linear terms or linear and quadratic terms etc. Secondly, a high degree of interaction necessitates a polynomial with a large number of terms, and this inevitably demands considerable time for each training run.

It can be seen from program A1 that the inner 'loops' numbered 45, 50 and 55 calculate the linear, quadratic and cubic terms, each with their respective coefficients. Loop 60 shifts the sub-data group through the history, and loop 35 selects the coefficient to be adjusted when these loops are completed. When each coefficient has been adjusted once, the entire process has to be repeated for the next run.

An important aspect of the use of a polynomial to represent a given length of history is the degree of interaction : this can readily be appreciated in the sciences of astronomy and meteorology; the motion and position of the stars are easily predictable, but the motion of cloud formations is not. A cine film of stars in motion can be shown backwards or forwards with no appreciable difference, but this could not be done with a film of moving cloud formations. One obvious reason for this as identified by Weiner [13] is that a linear time dependence exist for star movement but which does not exist for the movement of cloud formations.

In the universe there is a relatively small number of particles (planets) that are vast distances apart and thus loosely coupled, whereas atmospheric cloud formations can be considered as a large number of particles closely coupled. The differences thus stem from scaling and perceived interaction. It is clear that Newton's Laws of Motion can readily be applied to the first example which involves only a small degree of interaction, but for the second example, where this degree is much increased, the Laws are difficult to apply. This suggests that an alternative approach using a simplified model is required. The method used by Weiner in the development of his prediction theory is far from ideal and for time series with high interaction considers statistical distributions and probability distributions.

As an example of human learning, babies take approximately twelve months to learn to walk, even though evolution has given them the correct equipment. They still need many attempts using trial and error measurements to master the art of walking. It is not difficult to see that although digital computers are very fast in performing program functions they will still take more time to arrive at a particular solution using a learning process than by using a conventional matrix solving method. It is obvious that there are times when conventional methods do not apply or cannot be employed and the only way of obtaining a solution is by using a learning method. As a consequence, the learning method is used to give an insight into the underlying principles involved in a given learning process and results in a clearer understanding of the principles themselves and enabling more complex and involved learning programs to be used.

3.3 Theory of Proposed Method

The theory of prediction is based upon the simple but fundamental premise that nothing can happen without a cause. Thus, for a history of sufficient length and sampled at a high-enough rate to contain relative information, it is possible to predict all the future values of a function from selections of its history. In applying Kolmogorov's equation to this philosophy, it is necessary to use high order terms, but, at present, it is not clear what complexity is required for a particular waveform or history.

In establishing the relationship between the number of coefficients of the constructed polynomial and a given waveform, initial consideration is given to linear terms. In chapter 2 dealing with experimental results, it can be seen that various results are obtained for simple waveforms. In establishing a link between a given waveform and the number of coefficients needed to represent it, two approaches are feasible; firstly using spectrums and Fourier analysis, and secondly generating waveforms using a given set of coefficients and their corresponding starting values.

For a simple function exp (x) and the sequence $e^{0.1}, e^{0.3}, e^{0.5}$ etc. all that is necessary to step from any term to the next is to multiply the term by exp (INCREMENT) which, in this case, is exp (0.2): thus the above sequence can be represented by a starting value, exp (0.1), and one linear coefficient, exp (0.2), as in figure 3.3.1. For a sine wave, a minimum of two linear coefficients are needed C_2 is constant for all sine waves at - 1 and C_1 depends only on the delay and is $2 \cos \Delta$; thus C_1 is always < 2 for $\Delta > 0$.

A sinh function also needs a minimum of two linear coefficients: again C_2 is - 1 for all sinh functions but C_1 depends on the delay and is 2 cosh Δ; thus C_1 is always > 2 for $\Delta > 0$. For the ramp which also requires two linear coefficients, C_2 again equals - 1 but C_1 is 2; In simplistic terms, the process of counting 1, 2, 3, 4 etc. sequentially is not just achieved by adding 1 at each step; it may also be achieved by doubling the last number then subtracting the previous number. A common factor is thus required which links the sine, sinh and ramp functions, from which a method of analysis for determining the optimum number of linear coefficients may emerge. An exponential needs one linear coefficient, but from equations 3.3.1 and 3.3.2 below, it can be seen that sine and sinh functions have two exponential components.

$$\text{Sine } (x) = (e^{jx} - e^{-jx})/2j \qquad 3.3.1$$

$$\text{Sinh } (x) = (e^{x} - e^{-x})/2 \qquad 3.3.2$$

Thus it appears that for each exponential component of the waveform, a linear coefficient is needed.

It is also evident that the ramp is at the junction of the sine and sinh functions as the second coefficient for a sine function is < 2, for a ramp is = 2, and for a sinh function is > 2 (provided the delay is > 0). This is shown graphically in figure 3.3.2, and thus one may imply that the ramp lies on the boundary between exp (jx) and the exp (x) components. This also suggests that the coefficients can be used to clarify at least three distinct sets of waveforms: oscillatory (sine waves), linearly increasing (ramp) and exponentially divergent (sinh function). Conceptionally it is clear that this form of classification could be expanded upon and used for pattern recognition as the coefficients of a waveform hold relevant information concerning

the structure of that waveform. The coefficients however have to be associated with the corresponding starting values because the starting values contain information concerning the phase and amplitude of the waveform. This is evident for the case of any sine wave or cosine wave because if they are of the same frequency and amplitude, they will have the same coefficients, but with different starting values as the phases are different. A similar relationship exists between sinh and cosh functions.

It is clear that a DC level requires one coefficient, a ramp, as described previously, requires two coefficients, and an x^2 function requires three (1, -3 and 3). By arranging these coefficients in a particular pattern as shown in table 3.3.1, the result resembles a Pascal's triangle. As also shown in 3.3.1, the coefficients for a ramp predict any ramp of the form mx + c; similarly coefficients 1, -3,3 predict any function of the form $ax^2 + bx + c$. etc. It is evident that the coefficients shown in the triangle predict functions that occupy the boundary area between further classifications of waveforms.

The relationship between the exponential components in the waveform and the coefficients that represent them are shown in table 3.3.2 for three simple functions: $f(x) = \sum_{i}^{n} a_i e^{b_i x}$ for n = 1, 2 and 3. The way in which the coefficients change their values is shown in table 3.3.3: for $C_2 = -1$, pure sine, sinh and ramp functions exist. For $C_2 > -1$ or < -1, exponential components are introduced which increase and damp the amplitude respectively. In addition, the frequency of the function is also effected where C_2 varies from -1.

Where two ramps are added together, the product is another ramp: although the two waveforms each require two linear coefficients separately, when added together they always become a single ramp which, itself only requires two linear coefficients. For two sine waves of different frequencies, when added, the result is a waveform that requires four linear coefficients which, themselves, can be calculated from the coefficients of the separate waveforms. This is illustrated in the following case in which C_1 and C_2 are coefficients of one sine wave, and K_1 and K_2 are coefficients of another.

$$(1 - C_1^{\Delta} - C_2^{2\Delta})(1 - K_1^{\Delta} - K_2^{2\Delta})$$

$$= 1 - (K_1 + C_1)^{\Delta} - (K_2 + C_2 - C_1K_1)^{2\Delta} + (C_1K_2 + C_2K_1)^{3\Delta} + C_2K_2^{4\Delta} \qquad 3.3.3$$

The resulting waveform has coefficients of (-1), $(K_1 + C_1)$, $(K_2 - C_1K_1 + C_2)$, and finally $(- C_1K_2 - K_1C_2)$. In theory this can be expanded to cover any combination of any number of coefficients. If, for instance, $C_1 = K_1$ and $C_2 = K_2$, equation 3.3.3 becomes $(1 - C_1^{\Delta} - C_2^{2\Delta})^2$. Thus, only two linear coefficients are needed. This shows that any two waveforms with the same coefficients, when added, produce a waveform which requires only one set of the original coefficients : the additional information defining the resultant waveform is contained in the new starting values, which are themselves formed by adding the starting values of the original separate waveforms. This method used for combining waveforms, is based on a trial and error process which appears to be true for all situations. A similarity can thus be drawn between prediction and Z transformation where prediction is considered as a transformation from the time domain into the 'Kolmogorov' domain.

If a sine wave is oversampled, say, every degree, C_1 becomes $2 \cdot \cos(1)$, and, if the angle becomes smaller, C_1 approaches $2 \cos(0)$,

thus approaching, the value expected for a ramp. In cases of oversampling, difficulties can arise because four coefficients might be tried when only two are necessary. If more than an optimum (minimum) number of coefficients is used, the final values of the coefficients are reached via a slow convergence which itself implies additional time and storage. For a ramp, with, for example, coefficients of -1 and 2, if three coefficients are used, the solution becomes such that any one of the coefficients can be set to an arbitrary value, and the two remaining coefficients have values that are dependent upon it. This is illustrated for 3 coefficients and the data history 1 to 6:

$$\begin{aligned} 1C_3 + 2C_2 + 3C_1 &= 4 \\ 2C_3 + 3C_2 + 4C_1 &= 5 \\ 3C_3 + 4C_2 + 5C_1 &= 6 \end{aligned} \qquad 3.3.4$$

This resulting matrix after simplification gives :-

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} C_3 \\ C_2 \\ C_1 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix} \qquad 3.3.5$$

For $C_1 = \lambda$, $C_2 = 3 - 2\lambda$, and $C_3 = -(2 + \lambda)$, different solutions are obtainable if, say, C_2 is set to λ instead of C_1. These three solutions are three of an infinite number of solutions for each case since λ can be set to any value.

A problem that arises from using non-learning methods, eg. a direct matrix solution, is that although the same solution for a given input data set can be selected from the infinite number of solutions, a different input set of the same function gives a different solution. This gives a false impression that, because the coefficients are

different in value, the waveform has undergone some change. If, however, the correct number of coefficients are used, together with a learning method, it is clear that the waveform is completely represented by a set of constant coefficients.

The convergence rate for the program to reach an optimum solution is affected by the fact that each adjustment of C_1 affects the values of C_2 and C_3. If a number less than the optimum is used, the mean square error minimises; for the optimum number, however, the error continually reduces as training goes on, until it reaches the resolution allowed by the computer used.

The convergence rate is greatly affected by the chosen data, mainly because each consecutive adjustment slides the coefficients closer to the minimum mean square error. It is evident that the nearer to this minimum they start, the quicker it is reached. In the case of a ramp with data -1, 0, 1, and 2, only one adjustment is needed to set the coefficients to their optimum values. For data with a longer history, however, the trend is for increasingly slower convergence because of an increased number of calculations.

The waveforms which are shown in figure 3.3.3 and which are derived from the experimental results given in chapter 2 section 2.3 appear to disagree with the proposed idea of the coefficient-exponential relationship. It is known that for a waveform made up from two exponentials added together,

$$e^{nx} + e^{mx} = f(x) \qquad 3.3.6$$

where m and n are real and/or imaginery, the coefficients can be calculated as:

$$C_2 = - e^{n\Delta} e^{m\Delta} \qquad 3.3.7$$

and $$C_1 = e^{n\Delta} + e^{m\Delta} \qquad 3.3.8$$

where Δ is the sampling distance and is set to one. If C_2 and C_1 are set equal and to 0.4, as shown in figure 3.3.3a, e^n and e^m can be calculated as:

$$e^n = 0.8633\dot{} \qquad 3.3.9$$

and $$e^m = - 0.4633\dot{} \qquad 3.3.10$$

This implies that m is imaginery and, is of the forms q + pj, then f(x) becomes:

$$f(x) = e^{nx} + e^{(q + pj)x} \qquad 3.3.11$$

The function $e^{(q + pj)x}$ can be expanded to $e^{qx}(\cos px + j \sin px)$ which describes a dampled spiral, when plotted in real, imaginary and time axes. This illustrates the important factor that any one complex coefficient:

$$C_1 = a + jb \qquad 3.3.12$$

with a corresponding starting value $a_1 + jb_1$ can produce a spiral which can be plotted on real, imaginary and time axes, the spiral taking the form of

$$e^{ax}(\cos bx + j \sin bx) \qquad 3.3.13$$

as shown in figure 3.3.4a. If the waveform is viewed (figure 3.3.4b) such that only the real and time axes are seen, the spiral appears to be a cosine waveform : similarly, if only the imaginary and time axes are viewed the waveform (figure 3.3.4c) appears to be a sinewave.

The expansion of $e^{q + pjx}$, when substituted into equation 3.3.11 gives the function:

$$f(x) = e^{nx} + (\cos px + j \sin px)e^{qx} \qquad 3.3.14$$

which appears in the three dimensions as a damped spiral, but with an exponential in the real plane added to it. This is shown in figures 3.3.4 d and e for both real and imaginary axes against time.

A similar mathematical process can be applied to the other waveform as shown in figure 3.3.3b, where C_1 and C_2 are set equal and to 0.5. In this case, e^n and e^m can be calculated as:

$$e^n = 1 \qquad 3.3.15$$

$$e^m = -0.5 \qquad 3.3.16$$

This implies that n = 0 and, solving for m and substituting, equation 3.3.13 becomes:

$$f(x) = e^{-0.693x}(\cos 2\pi x + j \sin 2\pi x) \qquad 3.3.17$$

This represents a damped spiral displaced in the real plane by a d.c. level.

By applying the quadratic formula to equations 3.3.7 and 3.3.8 for the solutions of e^{mx} and e^{nx},

$$e^{mx}, e^{nx} = \frac{C_1 \pm \sqrt{C_1^2 + 4C_2}}{2} \qquad 3.3.18$$

For both m and n to be real,

$$C_1^2 + 4C_2 < C_1^2 \qquad 3.3.19$$

Therefore

$$4C_2 < 0 \qquad 3.3.20$$

$$C_2 < \frac{C_1^2}{4} \qquad 3.3.21$$

Thus, for the time series to be real, C_2 must be negative; if both coefficients are positive, waveforms with real and imaginary components can be generated.

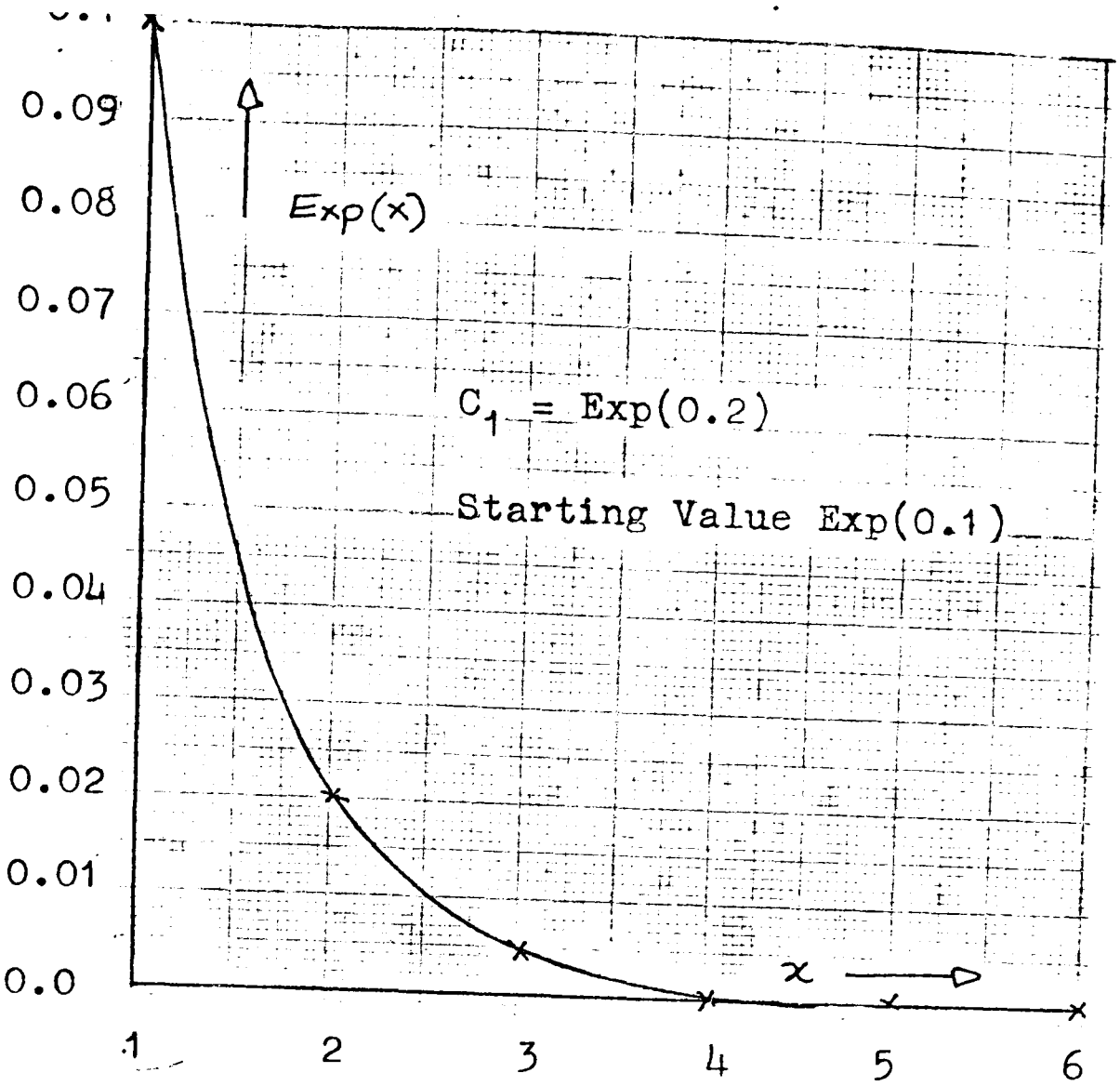


Figure 3.3.1 Waveform generated by one coefficient and one starting value.

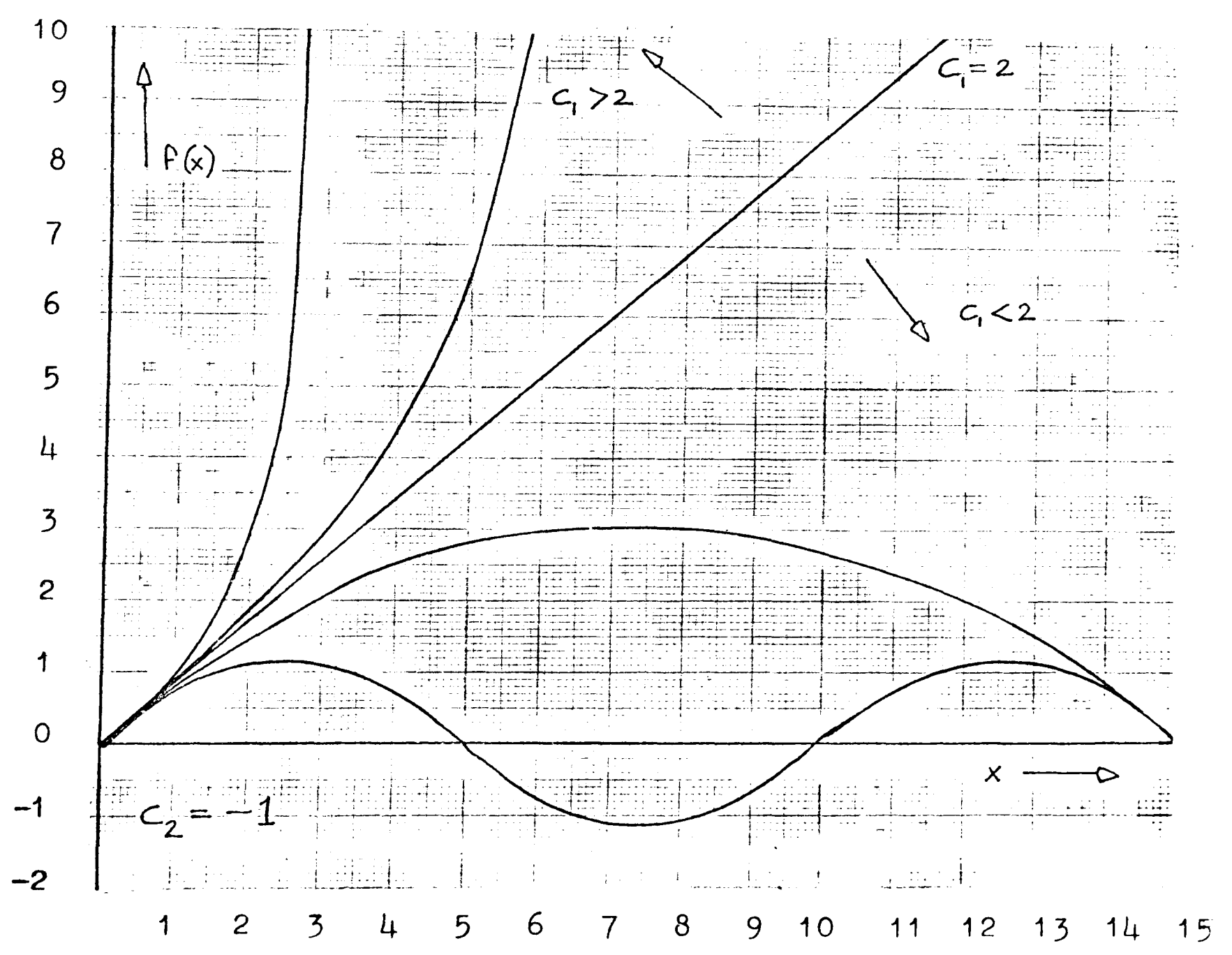
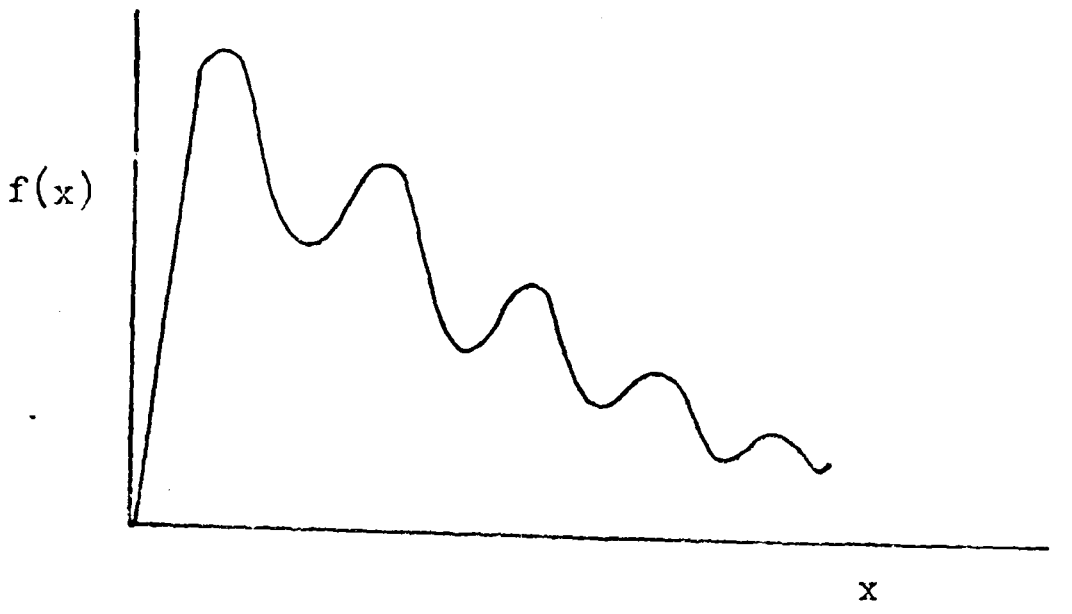
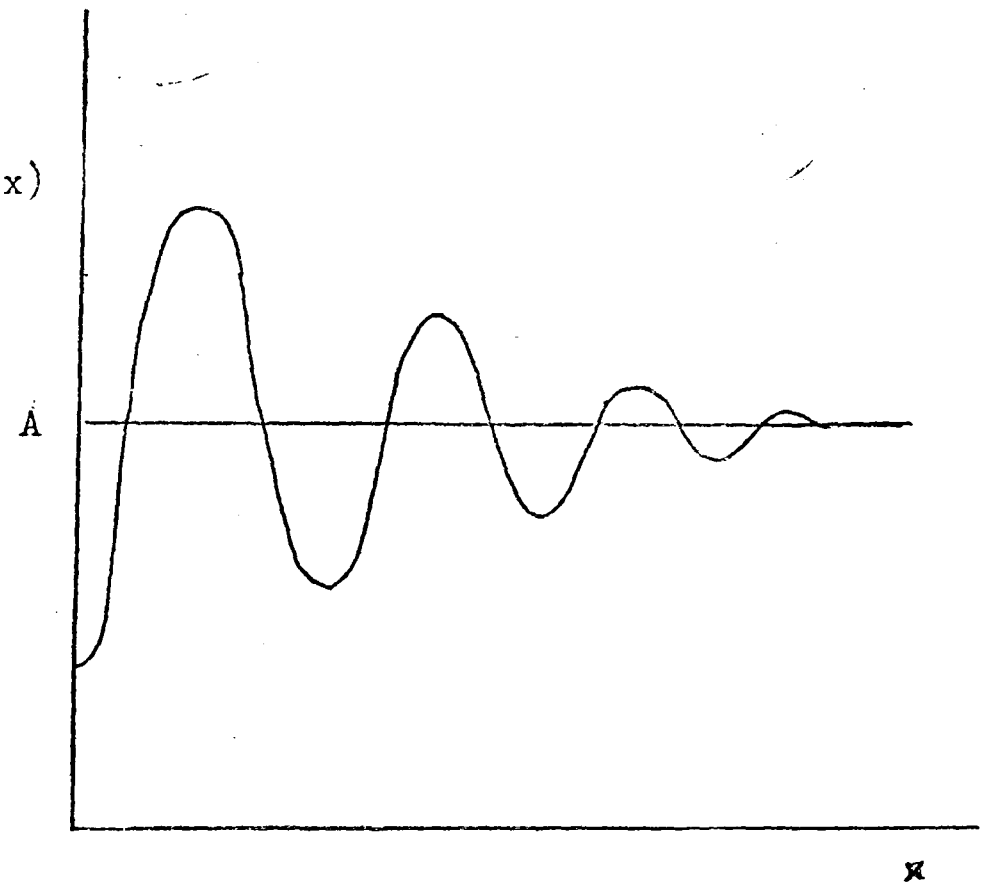


Figure 3.3.2 Relationship between coefficients and simple waveforms.



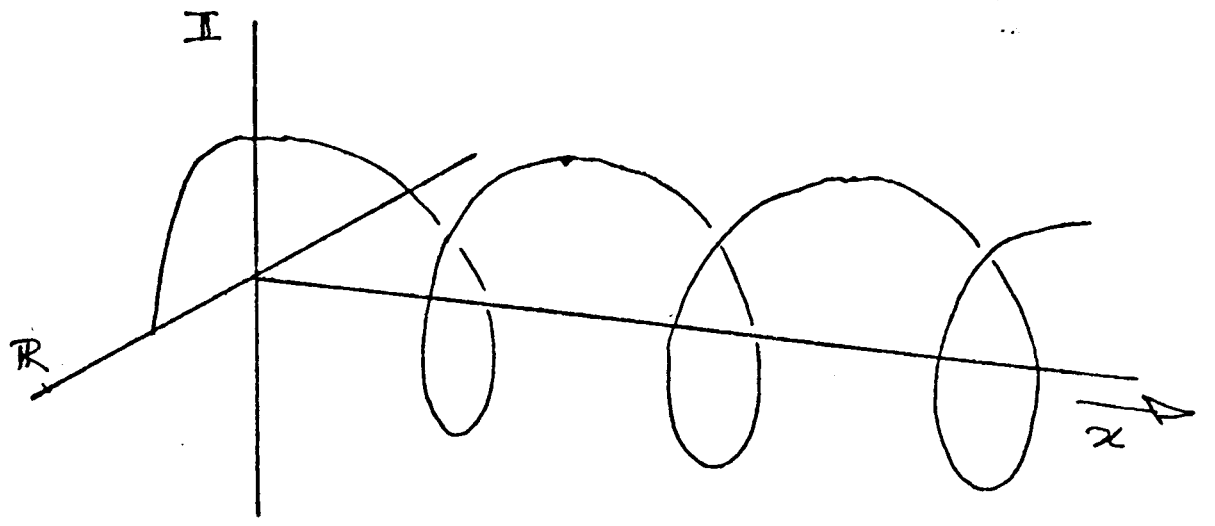
a) $f(x) = e^{nx} * \text{Sin}\ (mx) + 1^{px}$



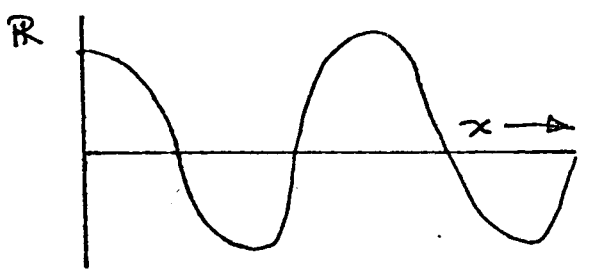
b) $f(n) = e^{nx} * \text{Sin}\ (mx) + A$

Figure 3.3.3 Two waveforms generated by two coefficients

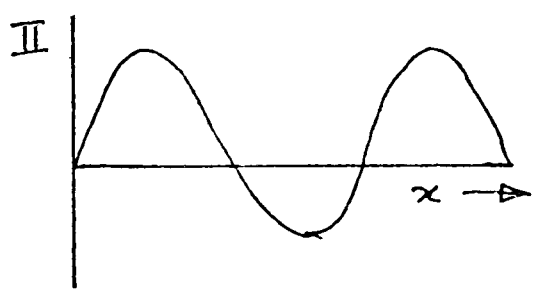
a)



b)

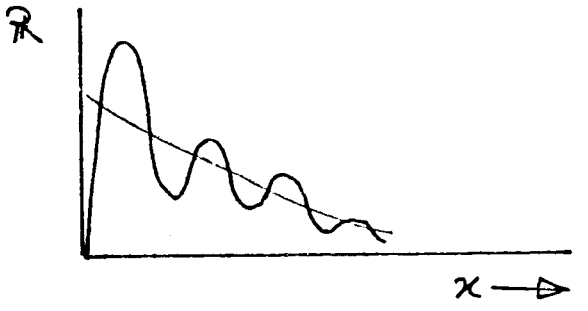


c)

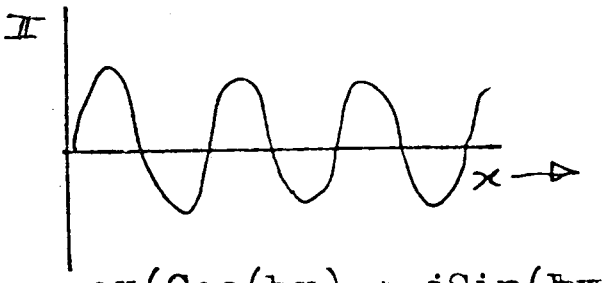


$C_1 = a + jb$, $f(x) = e^{ax}(Cos(bx) + jSin(bx))$

d)

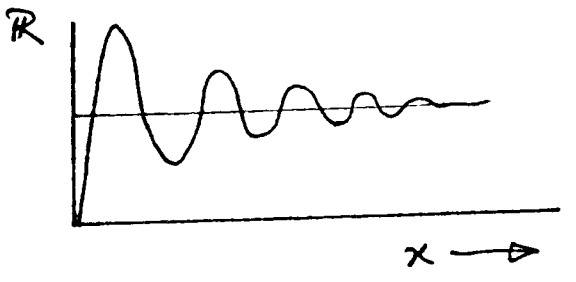


e)

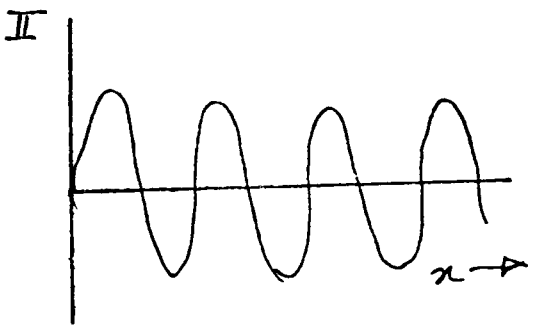


$C_1 = C_2 = 0.4$, $f(x) = e^{nx} + e^{ax}(Cos(bx) + jSin(bx))$

f)



g)



$C_1 = C_2 = 0.5$, $f(x) = A + B*e^{-0.693x}(Cos(2\pi x) + jSin(2\pi x))$

Figure 3.3.4 Waveforms generated by two real coefficients.

Coefficient Values	Relative Equations
1	A_0
-1 2 1	$B_1x + B_0$
1 -3 3 1	$C_2x^2 + x\ C_1x + C_0$
-1 4 -6 4 1	$D_3x^3 + x\ D_2x^2 + D_1x + D_0$
1 -5 10 10 5 1	$E_4x^4 + E_3x^3 + E_2x^2 + E_1x + E_0$
-1 6 -15 20 -15 6 1 etc.	etc.

Note: minus signs are ignored when generating the next line of coefficients.

Table 3.3.1 Pascal's coefficient triangle

f(x)	Coefficients
$f(x) = e^{mx}$	$C_1 = e^{m\Delta}$
$f(x) = e^{mx} + e^{nx}$	$C_1 = e^{m\Delta} + e^{n\Delta}$ $C_2 = -e^{m\Delta} * e^{n\Delta}$
$f(x) = e^{mx} + e^{nx} + e^{px}$	$C_1 = e^{m\Delta} + e^{n\Delta} + e^{p\Delta}$ $C_2 = e^{m\Delta} * e^{n\Delta} + e^{m\Delta} * e^{p\Delta} + e^{p\Delta} * e^{n\Delta}$ $C_3 = -e^{m\Delta} * e^{n\Delta} * e^{p\Delta}$

Table 3.3.2 Relationship between exponential components of the time domain waveforms

Function	Coefficients	
	C_1	C_2
Sin(x)	-1	2*Cos(△)
Ramp	-1	2
Sinh(x)	-1	2*Cosh(△)

Note: △ is the sampling interval.

Table 3.3.3. Relationship between simple functions and their coefficient's values.

3.4. Limitation of Proposed Method

Now a link has been established between the coefficients of a waveform and the exponential components of that waveform, bearing in mind that time is not of great importance. The information to be extracted does not in this case require real time analysis, as the data used can range over many years, or can be recorded data from a given system that requires to be simulated. If, however, speed is important, as in the case of speech encoding, where the coefficients, error signals etc. are needed for real time transmission, standard matrix manipulation methods can be used to obtain the optimum coefficients quickly, provided the machine has sufficient memory and the number of coefficients is known.

If the waveform can be assumed to fall into one of the classification for two linear coefficients:

1. Exponential increasing
2. Linearly increasing
3. Oscillatory
4. Damped oscillatory
5. Complex waveforms

different techniques have to be applied to certain category groups: if the waveform falls within the requirements of classes 3 or 4, the Fourier transform is easily implemented to show the number of exponential components contained in the waveform (figure 3.4.1 a to d) and, by their relative amplitudes, their respective importance. Waveforms within the remaining categories 1, 2 and 5 have characteristics which are best identified in the time domain.

For a category 3 oscillatory waveform for which a Fourier transform can be used with ease, the coefficients are readily related

to the lines in the power spectrum as shown in figure 3.4.2a. By associating the importance of a coefficient with the relative amplitude of its spectral line, the decrease in the mean square error for a given increase in the number of coefficients can be directly calculated. This is done by taking the spectrum and re-arranging the spectral lines in order of their amplitude and then calculating each amplitude as a percentage of the whole. In figure 3.4.2c, it is clear that the mean square error falls as the number of coefficients increases.

The standard practical prediction technique, given a section of waveform, is to calculate a set of coefficients which can be used with n sampled values as starting points to predict the waveform's future. Where a given waveform is represented by a set of n starting values and n constant coefficients, a deterministic function will always result because a set of constant coefficients and their corresponding starting values can only supply certain residual information concerning the structure of the waveform. (An analysis of this technique is given in section 3.3).

To accommodate the necessary flow of information in a non-deterministic waveform, the coefficients do not have to be constant. An alternate prediction technique, similar to a filtering or simulation process, can be used with a reduced number of varying coefficients whose values fluctuate with, and in proportion to, the information content of the given waveform. A comparison between the two techniques shows that a number of variable coefficients give a low average error, whilst a larger number of constant coefficients gives a slightly lower error. Thus, for a particular application, the choice of

technique type must depend upon the factors within that application. The waveform is divided into sets or groups, each of at least n points where n is the number of coefficients, and the coefficients can then be calculated as seen in the example of a square wave figure 3.4.3a and b for which the standard technique would try to fit a set of sinewave components, whereas the alternative method using one varying coefficient would produce an error of zero, except where the square wave crosses the time domain axis. Although the case of a square wave is unlikely to occur in practice, its analysis is useful as an example, particular as in this case, the single variable coefficient remains constant. A triangular waveform (figure 3.4.3 c and d) can similarly be considered in short groups as having two linear coefficients and appearing as a ramp function, except when the ramp changes slope and an error occurs.

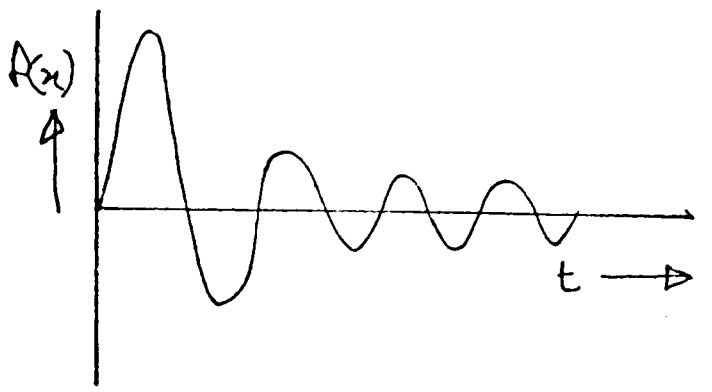
The waveforms used as examples are deterministic because they are mathematically predictable and have known components, whereas with general data, the interpretation placed upon results obtained could be biased as one could never be sure that the general data would not contain stochastic components that had not already been taken into consideration.

One of the major problems which can arise in prediction is that the available data may have previously been processed : this may or may not be advantageous and will affect the polynomial. One example which illustrates this concerns the Electronic Random Number Indicator Equipment (ERNIE) used to select winning premium bond numbers. If one were able to have access to the direct output of the machine

prediction of the selection process may be very much simplified compared with prediction based upon lists as they appear in the newspapers and which are already placed in alphanumeric order.

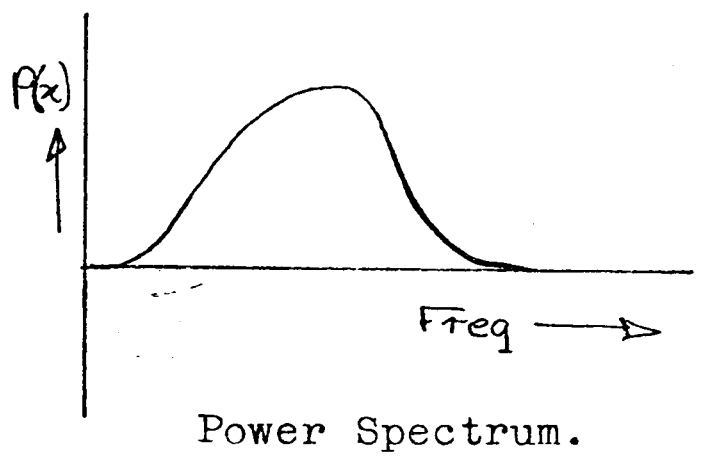
For data which is non-linearly sampled, minor processing has to be used to express it in linear form. The technique used employs the discrete Fourier transform, the principle being that a waveform of n points is transformed to a spectrum of n lines; thus, by extending this spectrum with zeros, (i.e. by padding it out, say, to m points and transforming back to the time domain) a waveform is obtained that exists in the same time interval but which has been interpolated and now has m points (where m is larger than n). Thus, to enable a waveform to be re-sampled at linear intervals, it is first necessary to define the required linear sampling period, and then examine the largest non-linear sampling period to enable the lowest common division to be identified for interpolation. The process is, however, limited to those waveforms which are transformable, and inevitably cannot be applied to waveforms which include exponentially increasing components.

a)



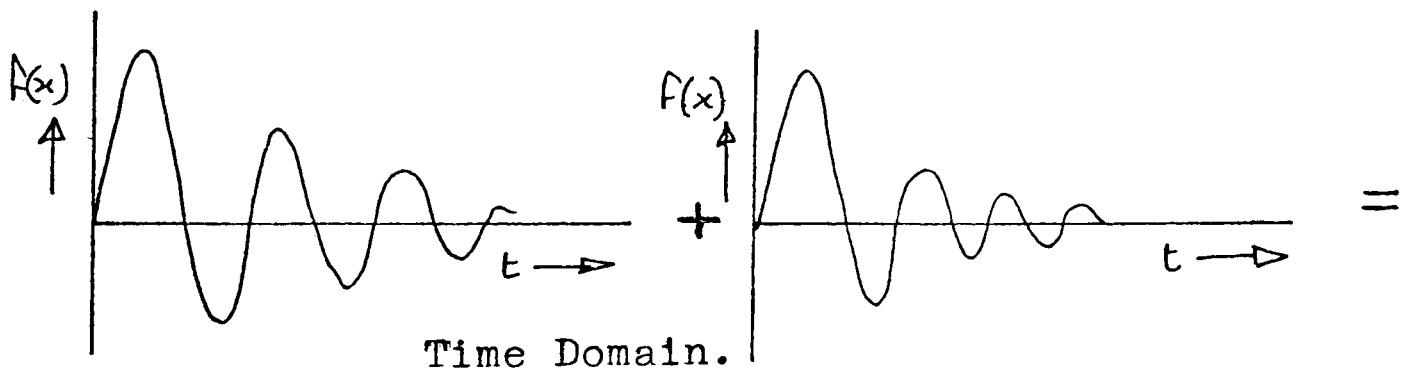
Time Domain.

b)



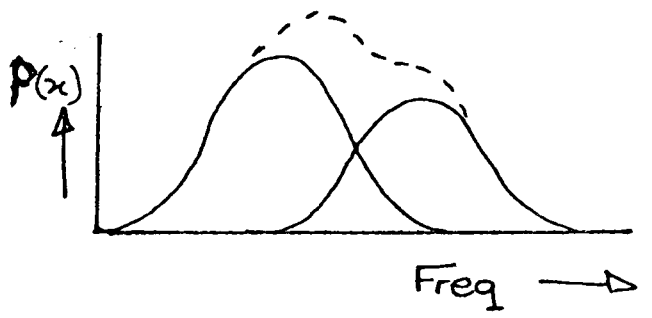
Power Spectrum.

c)



Time Domain.

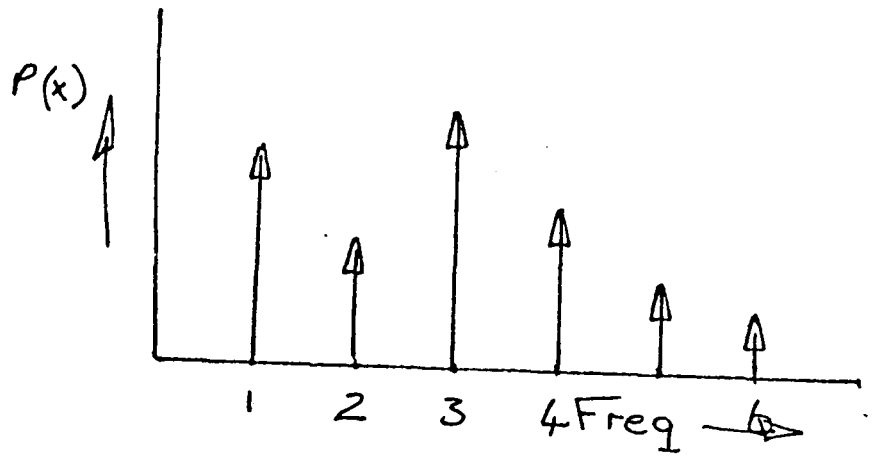
d)



Power Spectrum.

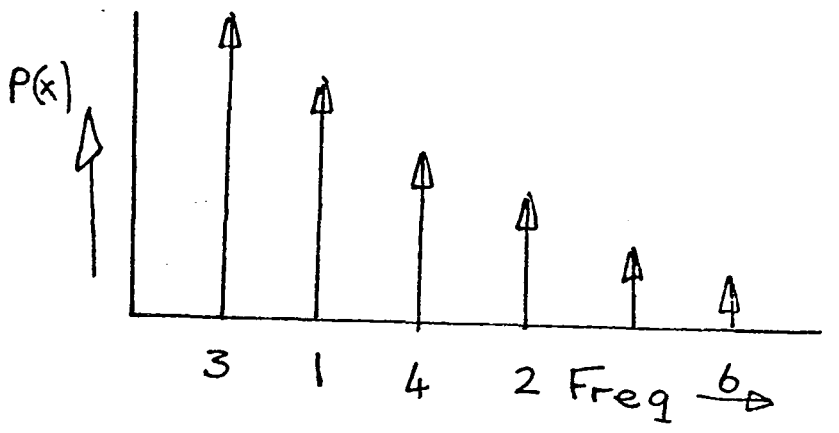
Figure 3.4.1 Representing Damped Sinewaves in the Frequency Domain.

a)



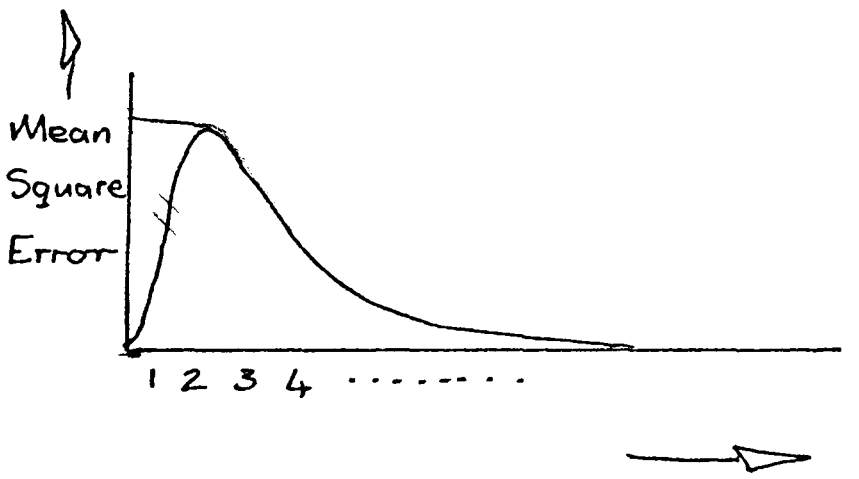
Power Spectrum.

b)



Rearranged Power Spectrum.

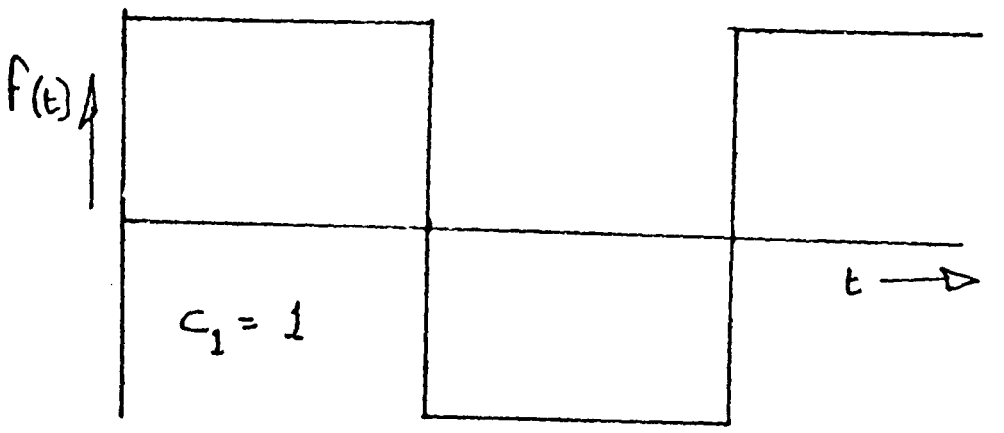
c)



Number of Coefficients.

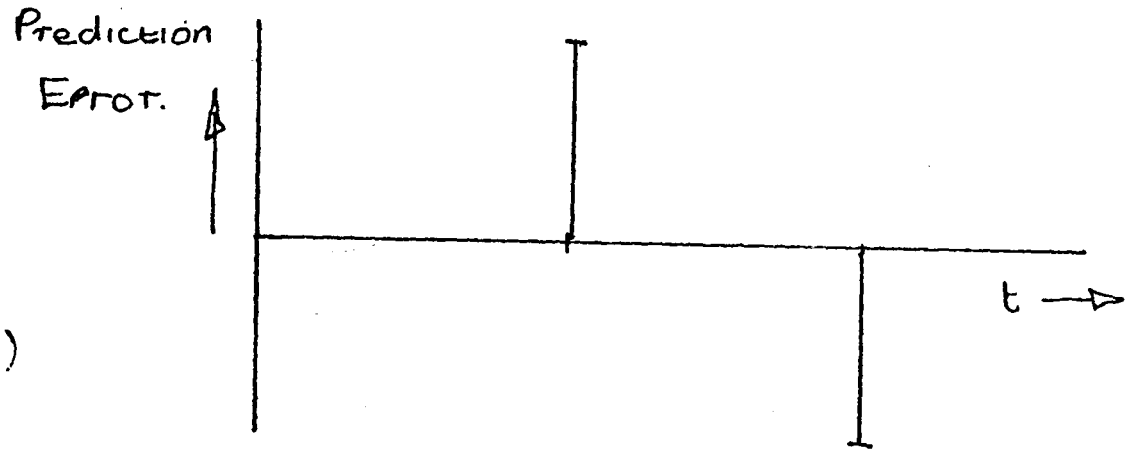
Figure 3.4.2 Rearrangement of Power Spectrum to produce variation of the Mean Square Error by increasing the number of coefficients.

a)



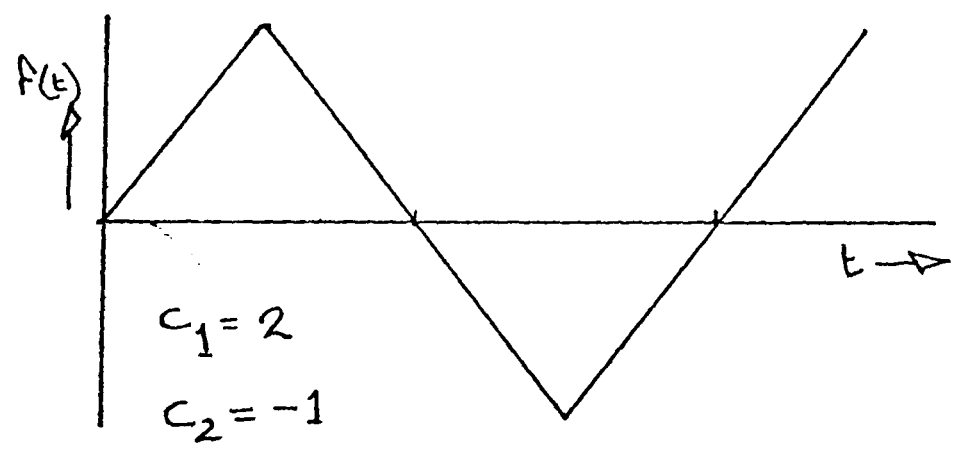
Time Domain.

b)



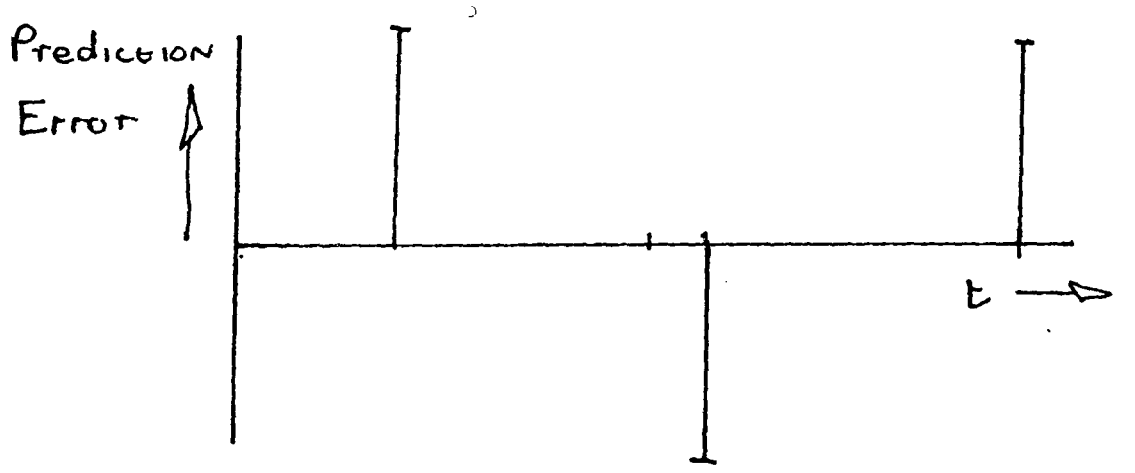
Prediction Error.

c)



Time Domain.

d)



Prediction Error.

Figure 3.4.3 Relationship between Prediction Error and the Time Domain Waveform.

3.5 A Non-Probabilistic Theory of Messages

The information flow in a perfect communication system as simplistically shown in figure 3.5.1 a passes unaltered and unhindered from transmitter to receiver. For a perfect imaging system, the invariant required needs to be geometric and independent of the object used or the way in which it is illuminated. The one and only invariant which fulfils this requirement is the Smith-Lagrange and which is defined as follows: the product of any small line element which is perpendicular to the optical axis and the angle of divergence of the rays which issue from any one of its points and which pass through a lens aperature is equal for both object and image (figure 3.5.1 b). This can be realised in a practical optic system provided certain lens errors are ignored and provided the element and divergence are restricted in size. For a simple transmitter as shown in figure 3.5.1 c, which consists of both a source and a coder, the information content of the output is exactly the same as that of the source, but is modified in presentation or appearance by the transform of the coder. In practical terms, this can be demonstrated by the use of a lens to produce the Fourier transform of a hologram.

Each point on a hologram (P_H) is the sum of the different intensities from each point on the object, as expressed in equation 3.5.1, and, conversely, each point on the object produces a different intensity at each point on the hologram. The intensities arriving at the hologram (as given in equation 3.5.2) are made up from the interference between the reference beam (A_r) and the object beam (A_o).

$$P_H = \sum_i^n W_i I_i \qquad 3.5.1$$

where: W_i is the weighting factor

I_i is the intensity

$$P_H = \sum_i^n \sum_j^n W_{ij} A_r(\lambda, d_j) A_o(\lambda, d_i) \qquad 3.5.2$$

where: A_r is the reference beam

A_o is the object beam

and, λ and d are parameters of A_r and A_o.

As interference is both in amplitude as well as phase, if the characteristics of the recording process are incorporated, the general formula can be re-expressed as equation 3.5.3, which is readily recognisable as being in the form of Kolmogorov's polynomial.

$$P_H = \sum_i W_i f(x,y)_i + \sum_i \sum_j W_{ij} f(x,y)_i f(x,y)_j + \ldots\ldots \qquad 3.5.3$$

where: W_i and W_{ij} are coefficients or weighting factors and determine the importance of the terms they are associated with, x and y are parameters of the function f(x,y), x can be time and y a predetermined delay.

the single summation encloses the linear terms.

the double summation encloses the quadratic terms ad infinitum.

Thus, if f(x,y) were a discrete time series with different delays, equation 3.5.3 could be expressed in the form

$$P = \sum_i^n C_i f(t - i\Delta) + \sum_i^n \sum_j^n C_{ij} f(t - i\Delta) f(t - j\Delta) + \ldots\ldots \qquad 3.5.4$$

P is the prediction value

where C_i and C_{ij} are coefficients

$f(t - i\Delta)$ is a function of time.

This equation was proposed by Fatmi, Nicholas and Young as an algorithm for intelligent-like machines.

The Kolmogorov polynomial will adapt itself (in its predictive mode) to represent systems varying in diversity from an industrial complex to the human vocal tract. If the coefficients remain constant, the system is of a deterministic nature and they represent the intrinsic information; conversely, if they vary with time, the system is stochastic in nature.

A laser can be used to produce a projection of a particular view of a hologram as shown in figure 3.5.2, and by scanning the laser, different views can be obtained. Each point on the hologram stores the total information with relationship to a particular view. It is thus reasonable to assume that by using the same polynomial or part of it, information extraction and storage should be possible. Because, however, of the vast amount of information stored in a hologram, a higher order polynomial is required than would be used for general information extraction from standard time series.

Any continuous variable such as speech or temperature variation, when sampled at the Nyquist rate, produce a discrete time series, which can be considered as defining a background or universe. When this background is 'viewed' through a window, the size of the window defines the number of combinations in the polynomial and, thus, the upper limit of the summation. If the window length is n, the linear terms in the polynomial can be expressed in the form

$$f(0) = \sum_{i}^{n} C_i f(t - i\Delta) \qquad 3.5.5$$

where $f(0)$ is some future value of $f(t - i\Delta)$
C_i the coefficients

Thus, the window length can be regarded as a lens that can transform the time series into a polynomial which can be used to represent the time series. By resolving the window length, the first step can be taken in calculating the number of terms needed in the polynomial.

As a hologram stores the total information of a scene and allows that same scene to be constructed at any later date, it may reasonably be assumed that by selecting sufficient terms of the Kolmogorov polynomial, it should be possible to store the total information of any signal in the coefficients. Each point on a hologram may be considered as a polynomial containing a selection of terms. The hologram itself, however, may be considered either as a set of polynomials in which each has a set of constant coefficients, or as one polynomial with time varying coefficients.

Once an appropriate number of terms in the polynomial has been chosen, the remaining problem is that of determining the values of the coefficients. For this, Wiener and Kolmogorov suggested minimisation of the mean square error :- if n coefficients (where n is the optimum number for a particular time series) are plotted in an n + 1 dimensional space (the extra dimension being the mean square error), the plot is an elliptical paraboloid with one absolute minimum, optimum solution. If, however, this optimum number of coefficients is increased, even by one, the coefficients are no longer independent and there are subsidiary minima. This can be readily illustrated using three linear coefficients on a ramp function, the result of which produces a plot that resembles a ploughed furrow; in this case, the furrow bottom forms an infinite set of minima, each of which is a solution. The optimum solution is that which sets the

extra arbitrary coefficient to zero. In practice, computer programs that use matrix methods to solve for the coefficients are presented with a problem, as there is an infinite set to choose from, and the solution is dependent upon the data set.

In the calculation of the coefficients, probabilistic concepts are not used. The coefficients are determined by simple arithmetic processes operating on the data points. Using the values of coefficients obtained by these simple processes, a polynomial is constructed which can adequately represent a sampled signal with a finite number of terms and which has a predetermined accuracy without having to resort to the use of probability distributions. Thus, the proposed theory of messages, based upon information concepts borrowed from physical optics, is non-probabilistic in all its essential concepts.

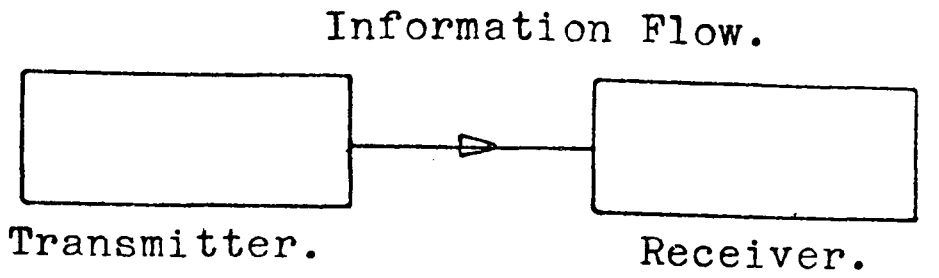


Figure 3.5.1a Information flow in a perfect communication system.

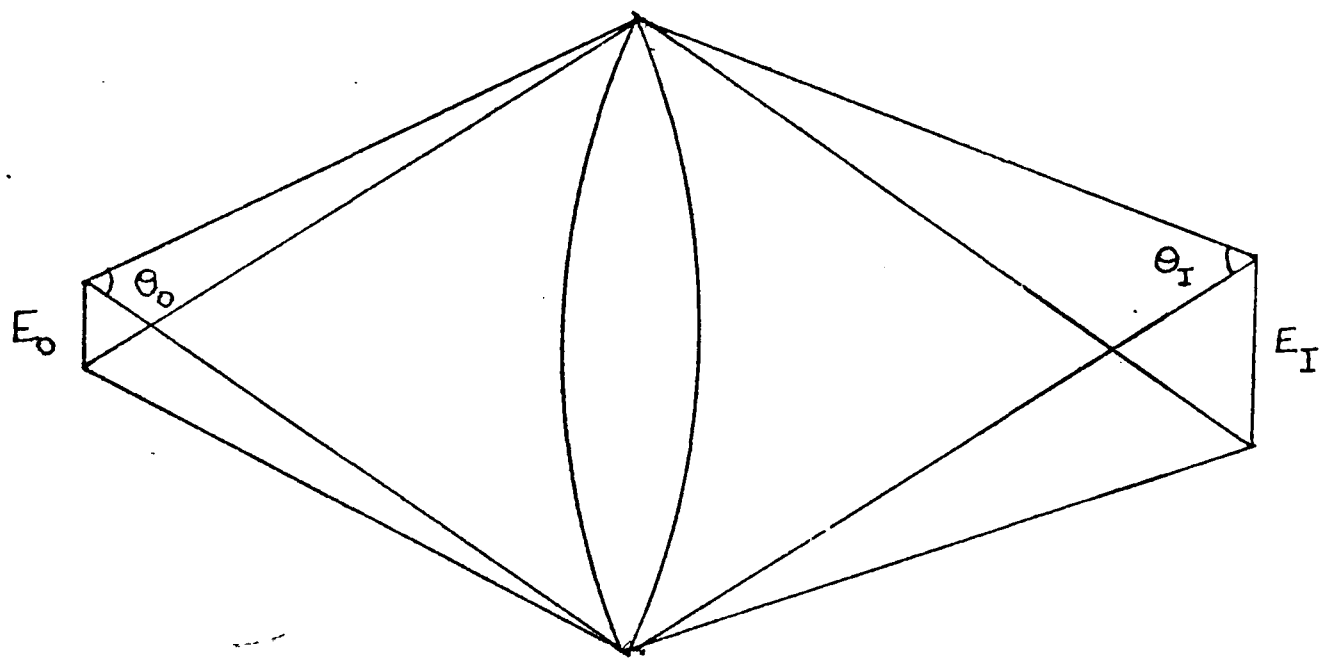


Figure 3.5.1b The 'Smith-Lagrange' invariant $E_0\theta_0 = E_I\theta_I$.

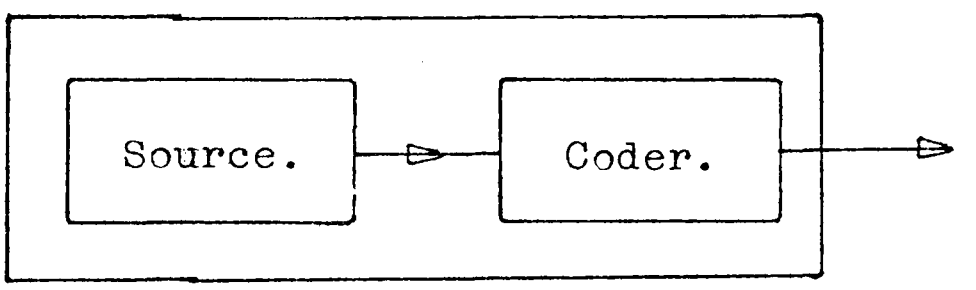


Figure 3.5.1c Source coding before transmission.

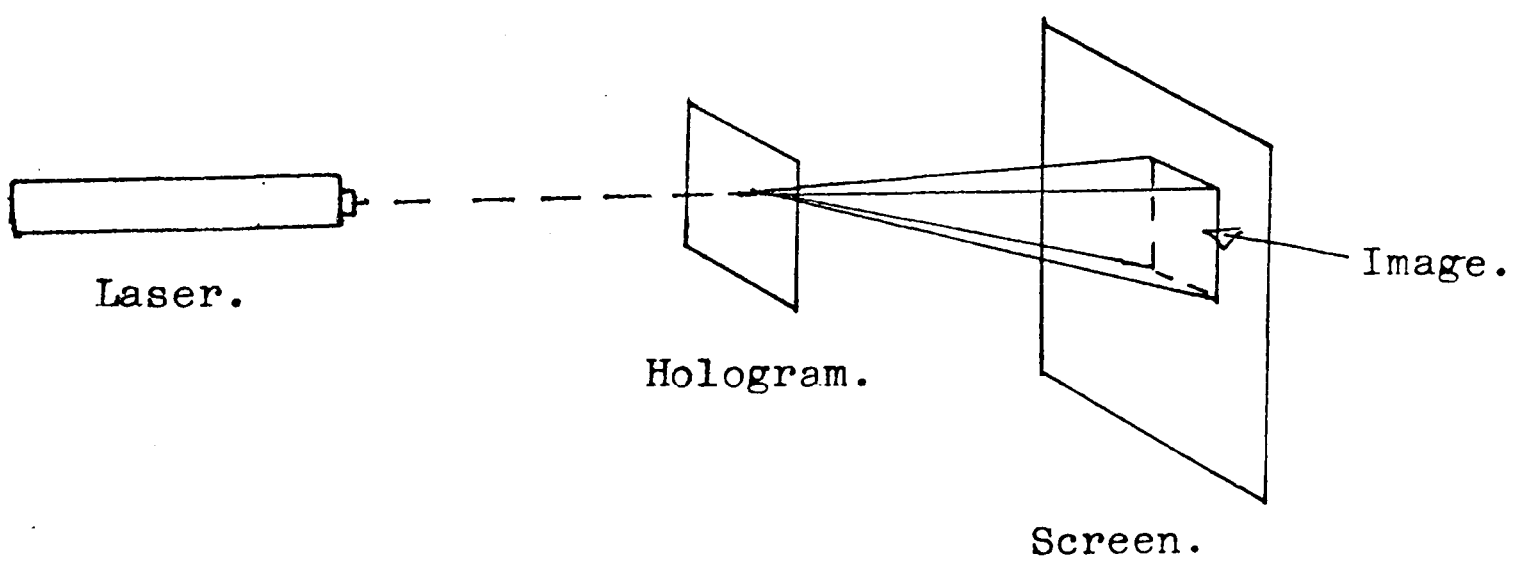


Figure 3.5.2 Projecting an image from a Hologram.

CHAPTER FOUR

4. CONCLUSIONS

4.1 Fast Kolmogorov transforms and information flow

Although the words Cybernetics and Prediction have been in the English language for some time, it is only comparatively recent that they have become common usage. In the past, however, words such as 'extrapolation' and 'forecast' have tended to be chosen by the media in preference to prediction and the like because of the unreasonable prejudice that prediction is associated with fortune telling and other similar dubious activities. Today, however, prediction as a word is used quite freely and may even be considered a vogue word. It is unfortunate, however, that such over-use has led to misuse, the users seldom realising that their application is just a minor aspect of prediction and is really only simple estimation.

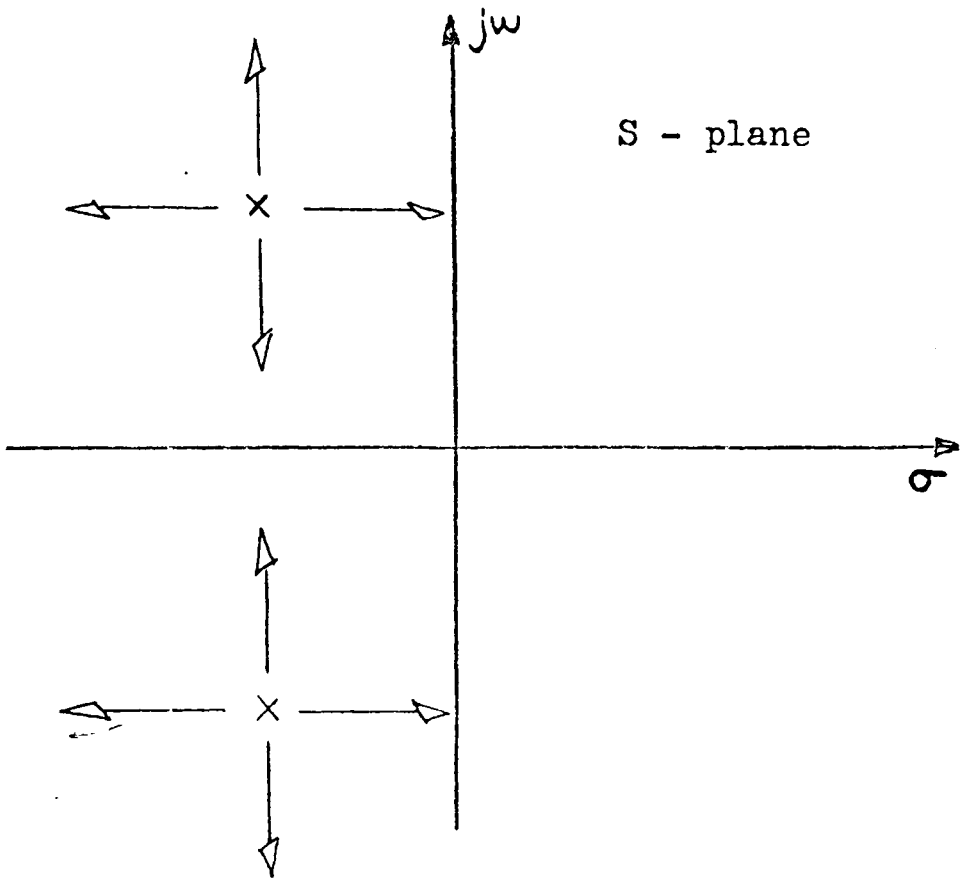
To facilitate prediction, information has to be extracted from the time domain waveform; a standard method is to use the Fourier transform which splits the waveform into its frequency components. The advent of high-speed digital computers has led to the development of discrete Fourier transforms, and by using the structure of the Fourier transform, faster versions have been developed which have made real-time fourier analysis a reality.

A link exists between prediction coefficients (and hence the information flow) and Laplace and Z transforms. An example of say, an exponentially decaying sine wave as shown in figure 4.1.1, illustrates the effects in the Laplace and Z domains by varying the

One coefficient can convert a sine wave into either an exponentially-increasing or decreasing sine wave if varied either side of minus one. If it is held at minus one, the other coefficient, when varied, affects the sine wave's frequency. Inevitably if both coefficients are varied together, their effects are interactive.

A possible future line of investigation is clearly that of the analysis and simulation of complex systems. If prediction techniques are used, the Z transform coefficients can be obtained directly from the time-domain waveform.

The methods used at present for information extraction employ matrix techniques to derive the information, and are given names such as Exact, Covariance and Autocorrelation (these are defined in detail in a report by Makhoul [42]). The Covariance method is perhaps the most widely used, and can be related directly to Kolmogorov's polynomial. For this reason, it is subsequently referred to as the Kolmogorov Transform. Using matrices techniques to remove redundancies and to arrange the elements of the matrix, the speed of obtaining the coefficients can be increased. This could produce a fast Kolmogorov transform resulting in the possible real-time analysis of the time domain waveforms. Again this is a very promising area for further investigation.



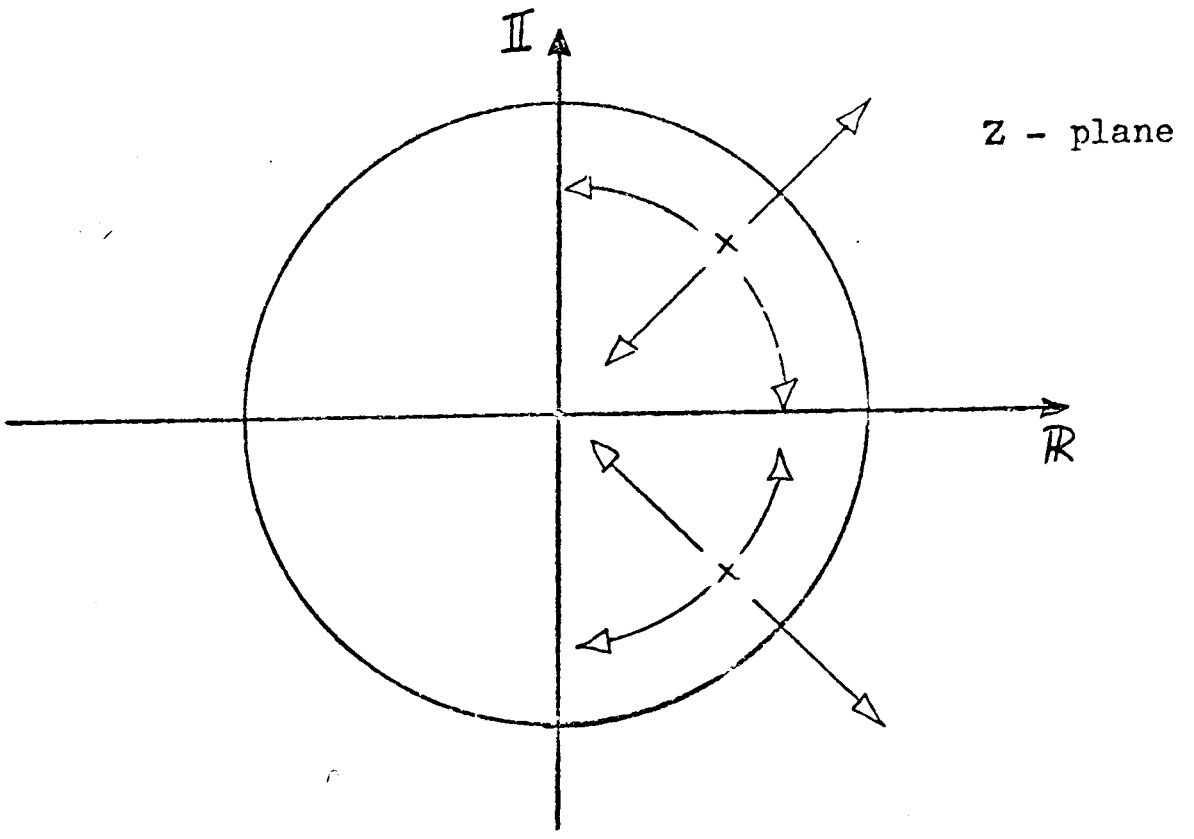


Figure 4.1.1 Effect of varying coefficients in the Laplace and Z Domains

4.2 Information flow and hyperspace

It is clear that the size of the Kolmogorov polynomial increases rapidly as it's ability to represent more complex systems increases. Thus, the ability of this polynomial to measure the information flow depends on the amount of information; the more information, the larger the polynomial.

If the concept of 'dimension' is considered in an abstract form and dissociated from any visual preconceptions, the information transmission and storage capabilities of speech are considerably inferior to the two-dimensional capabilities of vision, which, in turn, is vastly inferior to the three-dimensional capabilities of holograms. If this natural progression is theoretically continued and, at each stage, is related to Kolmogorov's polynomial, the result is a polynomial which represents information flow described in hyperspace. A truncation of this polynomial would reduce this hyperspace to an n-dimensional space which would be of more practical use, and which itself, would seem a promising area for research.

4.3 Structural aspect of information

It has been shown that the coefficients can be used to classify certain waveforms and it is suggested that this type of classification could be extended to include complex systems. Any system, be it a petrochemical works or a nuclear reactor, that can be considered as a black box with an input and an output can be simulated by the Kolmogorov polynomial. The simulator coefficients obtained could then be used to classify the system, the coefficients representing extracted information that relates to the size and structure of that system. An example where vast amounts of information not only flow through, but are retained in the structure, is the DNA molecule.

A suggested area for further investigation would be to examine the ability of the simulator coefficients to classify complex systems, and also to compare the coefficients from different systems with the relative size and structure of each system. In addition to this, the basic concepts of feedback, communication and control could be expanded in this context.

4.4 Information optics

An important relationship between holograms and Kolmogorov's polynomial has been developed. A hologram has the unique ability to store the total, three-dimensional information of a scene which can, therefore, be considered as either a group of polynomials with constant coefficients, or as a single polynomial whose coefficients vary with the flow of information. This variation could possibly be used to monitor the information flow, and would seem an ideal subject for further research. A suggested method would be to split the data into 'stationary' blocks and a set of n coefficients obtained for each block. This allows for a mean square difference calculation to be made for each coefficient, and would be a measure of the information flow. This method can also be used as a measure to show if the optimum number of coefficients is being used. By plotting the coefficients as they vary from block to block, their amount of variation decreases as the optimum number is approached. It is clear that if there is an exact number of coefficients for the data in question, the prediction error would be zero and the coefficients would not vary from block to block.

4.5 Final Conclusion

An account of an investigation into information flow in cybernetic systems has been presented. As a result of this investigation a theory of messages, which is non-probabilistic and which is based upon information concepts borrowed from physical optics, has been proposed.

The proposed theory brings together information flow and holograms and establishes a connection between the exponential components of a time domain waveform and the coefficients of the Kolmogorov polynomial.

PROGRAM A1

```
C      FORTRAN
C
       PROGRAM KOL (INPUT,OUTPUT)
C
C      PURPOSE
C           COMPUTER SIMULATION OF GABOR'S 'UNIVERSAL NON-LINEAR
C           FILTER, PREDICTOR AND SIMULATOR WHICH OPTIMIZES ITSELF
C           BY A LEARNING PROCESS PROC. IEE, VOL. 108, PART B,
C           NO. 40, JULY 1961'. WHICH WAS BASED ON KOLMOGOROV'S
C           POLYNOMIAL.
C
C      DEFINITION OF PARAMETERS
C           K          - LENGTH OF ARRAYS F AND T DIMENSION OF ARRAYS
                         CURRENTLY SET TO 20
C           F          - INPUT FUNCTION ARRAY
C           T          - TARGET FUNCTION ARRAY
C           NO         - NOISE DATA ARRAY
C           W          - ARRAY CONTAINING VALUE OF EACH TERM IN
C                        CONSTRUCTED POLYNOMIAL
C           C          - ARRAY CONTAINING CURRENT VALUES OF POLYNOMIAL
C                        COEFFICIENTS
C      NOTE:DIMENSION OF ARRAYS W AND C CURRENTLY SET TO 55 THIS
C           ALLOWS UP TO 5 LINEAR, 15 QUADRATIC AND 35 CUBIC TERMS.
C           AP         - ARRAY CONTAINING TERMS MULTIPLIED BY CORRESPONDING
C                        COEFFICIENTS DIMENSION OF ARRAY CURRENTLY SET TO
C                        55.
C           JI         - ARRAY CONTAINING SIZE OF SUBDATA GROUP WHEN MEAN
C                        SQUARE ERROR MINIMISED.
C           NI         - ARRAY CONTAINING DEGREE OF INTERACTION WHEN MEAN
C                        SQUARE ERROR MINIMISED
C           IU(1)      - NUMBER OF LINEAR TERMS IN POLYNOMIAL
C           IU(2)      - NUMBER OF QUADRATIC TERMS IN POLYNOMIAL
C           IU(3)      - NUMBER OF CUBIC TERMS IN POLYNOMIAL
C           IT         - TOTAL NUMBER OF TERMS IN POLYNOMIAL
C           ITA        - NUMBER OF TRAINING RUNS
C           EEA        - MEAN SQUARE ERROR
C
       DIMENSION F(20),T(20),NO(20),W(55),C(55),AP(55),JI(12),NI(12),IU(3)
C
C      READ LENGTH OF DATA SET AND MEAN SQUARE ERROR MINIMISATION LIMIT
C      TYPICALLY 10^-6 ALSO DEGREE OF INTERACTION AND MEAN SQUARE ERROR
C      LIMIT.
C
       READ 1100 K,AA,N,EL
 1100  FORMAT (I2,F10.0,I1,F.10.5)
       AAA = AA
C
C      READ SIGNAL, TARGET AND NOISE DATA
C
       READ 1200,(F(I),T(I),NO(I),I = 1,K)
 1200  FORMAT (3F10.5)
C
C      PRINT OUTPUT HEADER INFORMATION
C
       PRINT 150
  150  FORMAT (1H1,//,39X,* CONVERTING A NOISY SINE WAVE TO A PURE*,
```

```
      PRINT 200
  200 FORMAT (1HO,///,35X,* TRAINING STOPPED WHEN *)
      PRINT 250
  250 FORMAT (1H+,57X, * MEAN SQUARE ERROR WAS LESS THAN 0.1    *)
      PRINT 300
  300 FORMAT (1HO,///,46X,*J SIZE OF SUBDATA GROUP.*,
     1/,46X,*N DEGREE OF INTERACTION.*,
     2///,46X,*IU(1) NUMBER OF LINEAR TERMS IN POLYNOMIAL.*,
     3/,46X,*IU(2) NUMBER OF QUADRATIC TERMS IN POLYNOMIAL.*,
     4/,46X,*IU(3) NUMBER OF CUBIC TERMS IN POLYNOMIAL.*,
     5///,46X,*IT TOTAL NUMBER OF TERMS IN THE POLYNOMIAL.*,
     6/,46X*EEA MEAN SQUARE ERROR.*)
      PRINT 350, (NO(I), I = 1,K)
  350 FORMAT (1HO,//,1X,* NOISE FUNCTION DATA.*,//,13F10.5)
      PRINT 400, (T(I), I = 1,K)
  400 FORMAT (1HO,//,1X,* SINE FUNCTION DATA.*,//,13F10.5)
      PRINT 450, (F(I), I = 1,K)
  450 FORMAT (1HO,//,1X,* SINE PLUS NOISE DATA.*,//,13F10.5)
      PRINT 500
  500 FORMAT (1HO,*,J    N  IU(1)  IU(2)  IU(3)  IT    ITA        EEA*)
      PRINT 550
  550 FORMAT (1H+,52X,* COEFFICIENTS ARE :*)
      PRINT 600
  600 FORMAT (1H+,115X,* FORECAST VALUE :*)
C
C     SELECT WINDOW LENGTH AND CALCULATE NUMBER OF LINEAR TERMS
C
      N = 1
      DO 10 J = 2,5
      NM = 0
      IU(1) = J
      IU(2) = 0
      IU(3) = 0
      IF (N.EQ.1) GO TO 15
C
C     CALCULATE NUMBER OF QUADRATIC TERMS
C
  120 DO 20 IR = 1,J
      IU(2) = IU(2) + IR
   20 CONTINUE
      IF (N.EQ.2) GO TO 15
C
C     CALCULATE NUMBER OF CUBIC TERMS
C
      DO 25 NR = 1,J
      DO 25 MR = 1,NR
      IU(3) = IU(3) + MR
   25 CONTINUE
C
C     CALCULATE TOTAL NUMBER OF TERMS IN POLYNOMIAL
C
   15 IT = IU(1) + IU(2) + IU(3)
C
C     SET COEFFICIENTS AND TRAINING RUN COUNTER TO ZERO
C
      DO 30 JJ = 1,IT
      C(JJ) = 0.0
```

```
   30 CONTINUE
      ITA = 0
C
C     INCREMENT TRAINING RUN COUNTER
C
  110 ITA = ITA + 1
C
C     RESET MEAN SQUARE ERROR TO ZERO
C
      DO 35 IG = 1,IT
      EEA = 0.0
      EEB = 0.0
      EEC = 0.0
      EED = 0.0
C
C     CALCULATE LENGTH OF SUBDATA GROUP
C
      IRA = K - J
      KRB = - 1
C
C     SCAN SUBDATA GROUP THROUGH INPUT DATA
C
      DO 60 JK = 1,IRA
      IRB = JK + KRB
C
C     TRANSFER INPUT DATA TO ARRAY W
C
      DO 40 IRC = 1,J
      JRC = IRC + IRB
      W(IRC) = F(JRC)
   40 CONTINUE
      JR = 1
C
C     GENERATE LINEAR TERMS MULTIPLIED BY ASSOCIATED COEFFICIENTS
C     Z REQUIRED FOR MEAN SQUARE ERROR CALCULATION.
C
      DO 45 IAA = 1,J
      AP(JR) = W(IAA)*C(JR)
      IF (JR.EQ.IG) Z = W(IAA)
      JR = JR + 1
   45 CONTINUE
      IF (N.EQ.1) GO TO 70
C
C     GENERATE QUADRATIC TERMS MULTIPLIED BY ASSOCIATED COEFFICIENTS
C     Z REQUIRED FOR MEAN SQUARE ERROR CALCULATION.
C
      DO 50 IAB = 1,J
      DO 50 IAA = IAB,J
      AP(JR) = W(IAB)*W(IAA)*C(JR)
      IF(JR.EQ.IG) Z = W(IAB)*W(IAA)
      JR = JR + 1
   50 CONTINUE
      IF (N.EQ.2) GO TO 70
C
C     GENERATE CUBIC TERMS MULTIPLED BY ASSOCIATED COEFFICIENTS
C     Z REQUIRED FOR MEAN SQUARE ERROR CALCULATION.
C
```

```
      DO 55 IAC = 1,J
      DO 55 IAB = IAC,J
      DO 55 IAA = IAB,J
      AP(JR) = W(IAC)*W(IAB)*W(IAA)*C(JR)
      IF (JR.EQ.IG) Z = W(IAC)*W(IAB)*W(IAA)
      JR = JR + 1
   55 CONTINUE
C
C     SUMS ALL TERMS IN POLYNOMIAL AFTER RESETTING SUMMATION TO ZERO
C
   70 APA = 0.0
      DO 65 JR = 1,IT
      APA = APA + AP(JR)
   65 CONTINUE
C
C     CALCULATE ERRORS BETWEEN TARGET FUNCTION AND CONSTRUCTED
C     POLYNOMIAL WITH THE COEFFICIENT BEING TRAINED SET TO:
C     ITS CURRENT VALUE, 1,0 AND - 1 (THE LATTER THREE FORMING
C     ARBITRARY VALUES NEEDED TO CALCULATE LATEST OPTIMUM VALUE
C     OF COEFFICIENT BEING TRAINED)
C
      BP = APA - AP(IG)
      CP = BP - Z
      DP = BP + Z
      IK = (J + IRB + 1)
      TARGET = T(IK)
      EA = TARGET - APA
      EB = TARGET - BP
      EC = TARGET - CP
      ED = TARGET - DP
C
C     CALCULATE MEAN SQUARE ERRORS.
C     NOTE: ONLY EEA REQUIRES RATIONALIZING
C
      EEA = EEA + EA*EA/FLOAT(IRA)
      EEB = EEB + EB*EB
      EEC = EEC + EC*EC
      EED = EED + ED*ED
   60  CONTINUE
C
C     CALCULATE CURRENT OPTIMUM VALUE FOR COEFFICIENT BEING TRAINED
C
      C(IG) = (EEC - EED)/(2.0*((EEC + EED) - 2.0*EEB))
   35 CONTINUE
      JR = 1
C
C     CALCULATE FORECAST VALUE USING CURRENT VALUES OF COEFFICIENTS
C     AFTER RESETTING FRECAST VALUE TO ZERO
C
      FCAST = 0.0
C
C     GENERATE LINEAR TERMS
C
      DO 85 I1 = 1,J
      FCAST = FCAST + C(JR)*F(IRA + I1)
      JR = JR + 1
```

```
   85 CONTINUE
      IF (N.EQ.1) GO TO 90
C
C     ADD IN QUADRATIC TERMS
C
      DO 80 I1 = 1,J
      DO 80 I2 = I1,J
      FCAST = FCAST + C(JR)*F(IRA +I1)*F(IRA + I2)
      JR = JR + 1
   80 CONTINUE
      IF (N.EQ.2) GO TO 90
C
C     ADD IN CUBIC TERMS
C
      DO 75 I1 = 1,J
      DO 75 I2 = I1,J
      DO 75 I3 = I2,J
      FCAST = FCAST + C(JR)*F(IRA + I1)*F(IRA + I2)*F( IRA + I3)
      JR = JR + 1
   75 CONTINUE
C
C     CHECK LIMIT ON MEAN SQUARE ERROR
C
   90 IF (EEA.LT.EL) GO TO 95
C
C     CHECK IF MEAN SQUARE ERROR MINIMISED
C
      AAA = AAA + 0.0001 * EEA
      IF (EEA.GE.AAA) GO TO 105
      AAA = EEA
      GO TO 110
  105 NM = NM + 1
C
C     CHECK IF MEAN SQUARE ERROR HAS REACHED A MINIMUM ON THE LAST
C     FOUR CONSECUTIVE TRAINING RUNS
C
      IF (NM.EQ.10) GO TO 115
      AAA = EEA
      GO TO 110
  115 CONTINUE
C
C     STORE VALUES OF SIZE OF SUBDATA GROUP AND DEGREE OF INTERACTION
C     FOR WHICH MEAN SQUARE ERROR MINIMISED
C
      MN = MN + 1
      JI(MN) = J
      NI(MN) = N
C
C     RESET MEAN SQUARE ERROR MINIMISATION LIMIT
C
      AAA = AA
   95 CONTINUE
C
C     PRINT RESULTS
C
```

```
      PRINT 650,J,N,IU(1),IU(2),IU(3),IT,ITA,EEA
  650 FORMAT (1HO,1X,I1,3X,I1,2X,I2,5X,I2,5X,I2,4X,I2,3X,I4,4X,E11.4)
      PRINT 700, (C(IG),IG = 1,IT)
  700 FORMAT (1H ,52X,5F13.5)
      PRINT 750, FCAST
  750 FORMAT (1H+,120X,F10.5)
   10 CONTINUE
C
C     INCREMENT DEGREE OF INTERACTION AND REPEAT TRAINING FOR NEW
C     POLYNOMIAL IF LESS THAN OR EQUAL TO 3.
C
      N = N + 1
      IF (N.LE.3) GO TO 120
C
C     PRINT VALUES OF SIZE OF SUBDATA GROUP AND DEGREE OF INTERACTION
C     FOR WHICH MEAN SQUARE ERROR MINIMISED
C
      IF (MN.EQ.0) GO TO 125
      PRINT 800
  800 FORMAT (1H ,//,5X,* MEAN SQUARE ERROR MINIMISED FOR*)
      PRINT 850,(JI(I),NI(I),I = 1,MN)
      FORMAT (/,10X,*J = *,I2,5X,*N = *,I2)
  125 PRINT 900
  900 FORMAT (1HO,* RESULTS USING KOLMOGOROV POLYNOMIAL*)
      STOP
      END
```

PROGRAM A2

```
C      FORTRAN
C
       PROGRAM LTO (INPUT, OUTPUT)
C
C      PURPOSE
C           COMPUTER SIMULATION OF KOLMOGOROV'S POLYNOMIAL EMPLOYING
C           A LEARNING METHOD WHICH OPTIMISES THE POLYNOMIAL'S
C           COEFFICIENT VALUES BY MINIMISING THE MEAN SQUARE ERROR
C           BETWEEN THE POLYNOMIAL AND A DESIRED VALUE.
C           POLYNOMIAL TRUNCATED TO LINEAR TERMS ONLY.
C
C      DEFINITION OF PARAMETERS
C           N         - LENGTH OF ARRAYS T AND S DIMENSION OF ARRAYS
C                       CURRENTLY SET TO 100
C           S         - SIGNAL DATA ARRAY
C           T         - TARGET DATA ARRAY. FOR PREDICTOR T SET EQUAL
C                       TO S, FOR SIMULATOR S EQUALS INPUT DATA, T
C                       EQUALS OUTPUT DATA OF SYSTEM BEING SIMULATED.
C           C         - COEFFICIENT ARRAY, DIMENSION OF ARRAY
C                       CURRENTLY SET TO 20.
C           LW        - LENGTH OF SUBDATA GROUP (WINDOW LENGTH) EQUAL
C                       TO NUMBER OF LINEAR COEFFICIENTS IN POLYNOMIAL
C           IUL       - NUMBER OF SUBDATA GROUPS
C           ITR       - TRAINING RUN COUNTER
C
       DIMENSION S(100),T(100),C(20),TM(20),TC(20)
C
C      READ LENGTH OF DATA SET AND SUBDATA GROUP
C
       READ 100,N,LW
  100  FORMAT (I3,I2)
C
C      READ SIGNAL AND TARGET DATA
C
       READ 200, (S(I),T(I),I = 1,N)
  200  FORMAT (2F10.5)
C
C      CALCULATE NUMBER OF SUBDATA GROUPS
C
       IUL = N - LW
C
C      SET TRAINING RUN COUNTER TO ZERO
C
       ITR = 0.0
C
C      INCREMENT TRAINING RUN COUNTER
C
   10  ITR = ITR + 1
C
C      SELECT COEFFICIENT TO BE TRAINED
C
```

```
      DO 1 I = 1,LW
C
C     RESET MEAN SQUARE ERRORS TO ZERO
C
      SEP = 0.0
      SEN = 0.0
      SEM = 0.0
      SE = 0.0
C
C     SCAN SUBDATA GROUP THROUGH SIGNAL DATA
C
      DO 2 J = 1, IUL
C
C     RESET SUMMATED TERMS TO ZERO
C
      STM = 0.0
C
C     CALCULATE TERMS IN POLYNOMIAL
C
      DO 3 K = 1,LW
      TM(K) = C(K)*S(J + K - 1)
      TC(K) = S(J + K - 1)
C
C     SUM TERMS
C
    3 STM = STM + TM(K)
C
C     CALCULATE ERRORS
C
      E = T(J + LW) - STM
      EN = E + TM(I)
      EP = E + TM(I) - TC(I)
      EM = E + TM(I) + TC(I)
C
C     CALCULATE MEAN SQUARE ERRORS
C
      SEN = SEN + (EN*EN)
      SEM = SEM + (EM*EM)
      SEP = SEP + (EP*EP)
    2 SE = SE + (E*E)/FLOAT(IUL)
C
C     CALCULATE OPTIMUM COEFFICIENT VALUE
C
    1 C(I)=(SEM - SEP)/(2.0*(SEM + SEP - 2.0* SEN))
C
C     PRINT OPTIMISED COEFFICIENT VALUES
C
      PRINT 300, (C(I), I = 1,LW)
  300 FORMAT (10F10.0)
C
C     CHECK IF TRAINING RUN LIMIT REACHED
C
      IF (ITR.EQ.100) STOP
C
C     START NEXT TRAINING RUN
C
      GO TO 10
      STOP
      END
```

PROGRAM A3

```
C     FORTRAN
C
      PROGRAM SHKL (INPUT, OUTPUT)
C
C     PURPOSE
C          COMPUTER SIMULATION OF KOLMOGOROV'S POLYNOMIAL
C          EMPLOYING A LEARNING METHOD WHICH OPTIMISES THE
C          POLYNOMIAL'S COEFFICIENT VALUES BY MINIMISING THE
C          MEAN SQUARE ERROR BETWEEN THE POLYNOMIAL AND A
C          DESIRED VALUE. POLYNOMIAL TRUNCATED TO TWO LINEAR
C          TERMS SPECIFICALLY TO SHOW PROCESSES INVOLVED.
C
C     DEFINITION OF PARAMETERS
C          N         - LENGTH OF ARRAYS FV,T AND S DIMENSION OF
C                      ARRAYS CURRENTLY SET TO 20.
C          FV        - FORECAST VALUE ARRAY
C          S         - SIGNAL DATA ARRAY
C          T         - TARGET DATA ARRAY.  FOR PREDICTOR T SET
C                      EQUAL TO S, FOR SIMULATOR S EQUALS INPUT
C                      DATA, T EQUALS OUTPUT OF SYSTEM BEING
C                      SIMULATED.
C          C         - COEFFICIENT ARRAY, DIMENSION OF ARRAY
C                      EQUAL TO 2.
C          IRF       - NUMBER OF SUBDATA GROUPS (IE NUMBER OF
C                      DATA POINTS IS S - 2)
C          J         = TRAINING RUN COUNTER
C          IL        = TRAINING RUN LIMIT
C
      DIMENSION FV(20),T(20),S(20),C(2),A(60),B(60)
C
C     READ LENGTH OF DATA SET AND TRAINING RUN LIMIT
C
      READ 100,N,IL
  100 FORMAT (2I2)
C
C     READ SIGNAL AND TARGET DATA
C
      READ 200,(S(I),T(I),I = 1,N)
  200 FORMAT (2F10.5)
C
C     CALCULATE NUMBER OF SUBDATA GROUPS
C
      IRF = N - 2
C
C     SET TRAINING RUN COUNTER TO ZERO
C
      J = 0
C
C     INCREMENT TRAINING RUN COUNTER
C
```

```
   30 J = J + 1
C
C      SELECT COEFFICIENT TO BE TRAINED
C
       DO 10 I = 1,2
C
C      RESET MEAN SQUARE ERRORS TO ZERO
C
       SEP = 0.0
       SEN = 0.0
       SEM = 0.0
       SE = 0.0
C
C      SCAN SUBDATA GROUP THROUGH SIGNAL DATA
C
       DO 20 K = 1,IRF
C
C      CALCULATE FORECAST VALUE USING CURRENT COEFFICIENT VALUES
C
       FV(K + 2) = C(1)*S(K) + C(2)*S(K + 1)
C
C      CALCULATE ERROR BETWEEN TARGET SIGNAL AND FORECAST VALUE
C
       E = T(K + 2) - FV(K + 2)
C
C      CALCULATE MEAN SQUARE ERROR BETWEEN TARGET SIGNAL AND
C      FORECAST VALUE
C
       SE = SE + (E*E)/FLOAT(IRF)
       IF (I.EQ.2) GO TO 1
C
C      CALCULATE ERRORS BETWEEN TARGET SIGNAL AND POLYNOMIAL FOR
C      C(1) EQUAL TO 1,0 AND - 1
C
       EP = T(K + 2) - (S(K) + C(2)*S(K + 1))
       EN = T(K + 2) - (C(2)*S(K + 1))
       EM = T(K + 2) - (-S(K) + C(2)*S(K + 1))
       GO TO 2
C
C      CALCULATE ERRORS BETWEEN TARGET SIGNAL AND POLYNOMIAL FOR
C      C(2) EQUAL TO 1,0 AND - 1
C
     1 EP = T(K + 2) - (C(1)*S(K) + S(K + 1))
       EN = T(K + 2) - (C(1)*S(K))
       EM = T(K + 2) - (C(1)*S(K) -  S(K + 1))
     2 CONTINUE
C
C      CALCULATE MEAN SQUARE ERROR
C
       SEP = SEP + (EP*EP)
       SEN = SEN + (EN*EN)
       SEM = SEM + (EM*EM)
    20 CONTINUE
C
C      CALCULATE OPTIMUM COEFFICIENT VALUE
C
```

```
      C(I) = (SEM - SEP)/(2.0*((SEM + SEP) - 2.0* SEN))
   10 CONTINUE
C
C     STORE MEAN SQUARE ERROR AND LOG OF MEAN SQUARE ERROR
C
      B(J) = SE
      A(J) = ALOG10(SE)
C
C     CHECK IF TRAINING RUN LIMIT REACHED
C
      IF (J.LT.IL) GO TO 30
C
C     PRINT RESULTS
C
      PRINT 100
  100 FORMAT (1H,//////,10X,* RESULTS USING SHKL.*)
      PRINT 200,C(1),C(2)
  200 FORMAT (1H ,//,10X,*C(1) = *,F10.5,* C(2) = *,F10.5)
      PRINT 300
  300 FORMAT (///,10X,* DATA *)
      PRINT 400, (T(K), K = 1,N)
  400 FORMAT (/,10X,6F10.5)
      PRINT 500
  500 FORMAT (//,10X,* FORECAST*)
      PRINT 600, (FV(K),K = 1,N)
  600 FORMAT (/,10X,6F10.5)
      PRINT 700
  700 FORMAT (1H ,//,* TRAINING RUN *, 3X, * MEAN SQ.ERR.*,
     15X,*LOG M.S.E.*)
      PRINT 800
  800 FORMAT (1H ,/,60(/,48,I2,9X,E10.3,4X,F10.3))
      STOP
      END
```

BIBLIOGRAPHY

1. KOLMOGOROV, A.: Interpolation and Extrapolation of Stationary series, Bulletin de l'Academie des Sciences de l'URSS, Series Mathematiques, 1942, 2, p.3.
2. WEINER, N.: The Extrapolation, Interpolation and Smoothing of Stationary Time Series (Wiley, 1949).
3. WEINER, N.: Response of a Nonlinear Device to Noise, MIT Radiation Laboratory Report No. 129, 1942.
4. WEINER, N.: Nonlinear Problems in Random Theory (Wiley, 1958).
5. BOSE, A.G.: A Theory of Nonlinear Systems, MIT Research Laboratory of Electronics Report No. 309, 1956.
6. ZADEH, L.A.: and RAGAZZANI, J.R.: An Extension of Weiner's Theory of Prediction, ibid., 1950, 21. p. 645. Optimum Filters for the Detection of Signals in Noise. Proc. IRE, 1952, 40, p.1223.
7. SINGLETON, H.E.: Nonlinear Systems with Sampled Input, MIT Research Laboratory of Electronics Report No. 160, 1951.
8. WHITE, W.D.: The Role of Nonlinear Filters in Electronics Systems, Proc. National Electronics Conference, 1953, 9, p.505.
9. BODE, H.W.: and SHANNON, C.E.: A Simplified Derivation of Linear Least Square Smoothing and Prediction Theory, Proc. IRE, 1950 (April), p.417.
10. SHANNON, C.E.: A Mathematical Theory of Communication, Bell Syst. Tech. J., vol. 27, p.379, July 1948, also p.623, Oct. 1948.
11. PIERCE, J.R.: The early days of Information Theory. IEEE Trans. Information Theory, vol. IT-19, p.3, Jan. 1973.
12. SLEPIAN, D.: Information Theory in the Fifties, IEEE Trans Information Theory, vol. IT-19, p.145, March, 1973.

13. WEINER, N.: CYBERNETICS: Control and Communication in the Animal and the Machine, MIT press, 1961.

14. GABOR, D.: Light and Information Ritchie Lecture, University of Edinburgh, March 2, 1951.

15. EDDINGTON, A.: The Philosophy of Physical Sciences, 1939, Cambridge, p.16.

16. GABOR, D.: Theory of Communication, Jour IEE, vol. 93, part 3, no. 26. p.429, 1946.

17. NYQUIST, H.: Certain topics in Telegraph Transmission Theory, AIEE Trans, p.617, April 1928.

18. LEITH, E.N.: Dennis Gabor, Holography and the Nobel Prize, Proc IEEE, vol. 60, No. 6, June 1972.

19. GABOR, D.: A Universal Non-Linear, Predictor and Simulator which optimizes itself by a Learning Process, Proc. IEE, vol. 108, part B, no. 40, July 1961.

20. YOUNG, R.: A Universal Machine, Chelsea Report, 1971.

21. MUFTOGULU, M.: Cybernetic Model of Hydrological System PhD. Thesis, Chelsea College, 1972.

22. KREIN, M.: On a problem of Extrapolation of A. N. Kolmogoroff, Comptes Rendus (Dokladz) de l'Academie des Sciences de l'URSS, 1945, vol. XLVI, no. 8, p.306, 1945.

23. YAGLOM, A.M.: Outline of some topics in Linear Extrapolation of Stationary Random Processes, Fifth Berkeley Symposium.

24. YAGLOM, A.M.: Extrapolation, Interpolation and Filtration of Stationary Random Processes, with Rational Spectral Density, Trudy Moskov. Mat. Obsc. 4 (1955), p.333.

25. BROWN, J.L.: On the Error in Reconstructing a Non-Bandlimited Function by Means of a Bandpass Sampling Theory, Jour Math. Anal. Appl. vol.18, no. 1, April 1967, p.75.

26. BROWN, J.L.: Anharmonic Approximation and Bandlimited Signals, IEEE Information and Control, vol.10, no. 4, Apr. 1967, p.409.

27. BROWN, J.L.: Truncation Error for Band-Limited Random Processes, Information Sciences, vol. 1, no.3, July 1969, p.261.

28. BROWN, J.L.: Bounds for Truncation Error in Sampling Expansions of Band-Limited Signals, IEEE Trans. Information Theory, vol. IT-15, no.4, July 1969, p.440.

29. BROWN, J.L.: Uniform Linear Prediction of Bandlimited Processes from Past Samples, IEEE Trans. Information Theory, Sept. 1972.

30. HAJEK, J.: Predicting a Stationary Process when the correlation Function is convex, Czechoslovak Math. j., vol.8, 1958, p.150.

31. DAVISSON, L.D.: Steady-State Error in Adaptive Mean-Square Minimization, IEEE Trans Information Theory, vol. IT-16, no. 4, July 1970.

32. GRAFAREND, E. and KELM, R.: Point and Inverval Estimations Especially of Point Errors, in Multidimensional Lease-Squares Adjustment, Bull. Geod, no. 104, July 1972, p.165.

33. GARUDACHAR, B.: Optimal Linearization Technique for Second-Order Non-Linear Equations, IEE - IERE Proc - India, March - April 1972, p.34.

34. GULYAS, O.: On Extended Potential Function Type Learning Algorithms and their Convergence Rate, Problems of Control and Information Theory, vol.1(1), 1972, p.51.

35. AINSWORTH, W.A.: On the Efficiency of Learning Machines, IEEE Trans. Syst. Sci. and Cyb., Nov 1967, p.111.

36. BOHLIN, T.: Comparison of two methods of Modeling Stationary EEG Signals, IBM J. Res. Dev., May 1973, p.194.

37. FENWICK, P.B.C., MICHIE, P., DOLLIMORE, J. and FENTON, G.W.: Mathematical Simulation of the Electroencephalorgram using an Autoregressive Series Bio-Med. Comput., vol. 2, 1971, p.281.

38. GERSCH. W.: Spectral Analysis of EEG's by Autoregrssive Decomposition of Time Series, Math. Biosci, vol.7, 1970, p.205.

39. WENNBERG, A and ZETTERBERG, L.H.: Application of a Computer Based Model for EEG Signals, Electroencephologr. Clin. Neurophys. vol, 31, no. 5, 1971 p.457.

40. ROBINSON, E.A.: Predictive Decomposition of Time Series with Application to Seismic Exploration, Geophysics, vol. 32, no. 3, June 1967, p.418.

41. ROBINSON, E.A. and TREITEL, S.: Introduction, Special Issue on the MIT Geophysical Analysis Group Reports, Geophysics, vol. 32, no. 3, June 1967, p.416.

42. MAKHOUL, J.I. and WOLF, J.J.: Linear Prediction and the Spectral Analysis of Speech, Report No. 2304, Bolt, Beranek and Newman Inc., Cambridge, Mass. (Aug. 1972).

43. ROBINSON, A.H. and CHERRY, C.: Results of a Prototype Television Bandwidth Compression Scheme, Proc. IEEE, vol. 55, no. 3, March 1967, p.356.

44. ELIAS, P.: Predictive Coding Part I and II, IRE Trans. Information Theory, vol IT-1, March 1955, p.16.

45. ATAL, B.S. and SCHROEDER, M.R.: Predictive Coding of Speech Signals, Wescon Tech. Papers, Paper 8/2, May 1968.

46. SCIULLI, J.A. and CAMPANELLA, S.J.: A Speech Predictive Encoding Communication System for Multichannel Telephony, IEEE Trans on Comm. vol-com-21, no. 7, July 1973, p.827.

47. KOBAYASHI, H. and BAHL, L.R.: Image Data Compression by Predictive Coding I: Prediction Algorithm, IBM J. Res. Develop., March 1974, p.164.

48. DAVISSON, L.D.: Data Compression Using Straight Line Interpolation, IEEE Trans. on Information Theory, vol IT-14, no. 3, May 1968, p.390.

49. DAVISSON, L.D.: An Approximate Theory of Prediction for Data Compression, IEEE Trans. on Information Theory, vol IT-13, no. 2, April, 1967, p.274.

50. DAVISSON, L.D.: The Theoretical Analysis of Data Compression Systems, Proc. IEEE, vol. 56, no. 2, Feb. 1968, p.175.

51. ANDREWS, H.C.: A Generalised Technique for Spectral Analysis, IEEE Trans. on Computers, vol C-19, no. 1, January 1970 p.16.

52. PRATT, W.K. KANE, J. and ANDREWS, H.C.: Hadamard Transform Image Coding, Proc. IEEE, vol. 57, January 1969, p.58.

53. ALGAZI, V.R. and SAKRISON, D.J.: On the Optimality of the Karhunan - Loeve Expansion, IEEE Trans on Information Theory, March 1969, p.319.

54. Special Issue on Fast Fourier Transform, IEEE Trans. Audio and Electroacoustics, vol AU-17, June 1969, p.25.

55. COCHRAN, W.H. et al,: What is the Fast Fourier Transform?, IEEE Trans. Computer, vol C-17, April 1968 p.373.

56. ANDREWS, H.C.: A high-speed Algorithm for the computer Generation of Fourier Transforms, IEEE Trans. Computers, vol C-17, April 1968, p.373.

57. FANO, R.M.: Short-time autocorrelation Functions and Power Spectra, Jour. Acoust. Soc. Amer. vol. 22, no. 5, Sept. 1950, p. 546.

58. MAKHOUL, J.: Spectral Analysis of Speech by Linear Prediction, IEEE Trans. on Audio and Electroacoustics. vol. AU-21, no. 3, June, 1973, p. 140.

59. ATAL, B.S.: Characterisation of Speech Signals by Linear Prediction of the Speech Wave, Proc. IEEE Symp. on Feature and Selection in Pattern Recognition, Argonne, IU. Oct. 1970, p.202.

60. ATAL, B.S. and HANAUER, S.L.: Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. Jour. of the Acoustical Soc. Am., vol. 50, no. 2 (part 2), August 1971, p. 637.

61. CLIFTON, K and FATMI, H.: Information Flow in Learning and Adaptive Systems, Biokybernetic IV, Leipzig, 1973.

62. CLIFTON, K. and FATMI, H.: A non-probabilistic theory of messages based upon information optics. VIII. Internat. Congr. Cybernetics, Namur 1973.

63. CLIFTON, K. and FATMI, H.: Information Flow in Cybernetic Systems. 3rd. Ann. Conf. 'Recent Topics in Cybernetics' Chelsea College, Sept. 1975.